

# Memoria del proyecto Baleike.com

**Autor: Gorka Julio**

**Empresa: Elurnet Informatika Zerbitzuak S.L.**

# Presentación

El proyecto es un proyecto empresarial que consisten en la migración, actualización y replanteamiento de un medio de comunicación digital Baleike.com. El cliente se pone en contacto con nosotros con la intención de actualizar su web y realizar una consultoría sobre las mejores opciones para ello.

El medio de comunicación tiene sobre todo una web <http://baleike.com> , pero también una comunidad de blogs asociada (<http://baleike.com/blogak/>). Junto con diferentes perfiles en diferentes servicios web. En la actualidad se encuentran con varios problemas que quisieran solucionar integrando un CMS que necesariamente sea Software Libre con los diferentes servicios web que utilizan. Teniendo en cuenta siempre o en la mayor medida de lo posible la independencia sobre estos, ya que en algún momento podrían fallar.



Ilustración 1: El portal anterior de baleike.com

El va desde la consultoría, hasta la implantación, trata la migración de blogs, pero sobre todo el desarrollo de la nueva web baleike.com. Un medio de comunicación creado utilizando el CMS Wordpress, integrando y desarrollando diferentes módulos (plugins) que le permiten conseguir las funcionalidades necesarias para este medio de comunicación. Parte importante del proyecto consiste en la interacción entre la plataforma y servicios web para la identificación y obtención de recursos.

# Tabla de contenidos

Presentación.....	2
1. Estudio de viabilidad.....	6
1.1. Establecimiento del alcance del sistema.....	6
Objetivos detallados.....	7
baleike.com.....	7
baleike.com/blogak/.....	8
1.2. Estudio de la situación actual.....	9
Identificación de los sistemas actuales .....	9
Descripción de los sistemas actuales .....	10
Baleike.com.....	10
Balieke.com/blogak.....	11
Diagnóstico de los sistemas actuales .....	11
Baleike.com.....	11
Baleike.com/blogak.....	12
1.3. Definición de requisitos del sistema.....	12
Requisitos técnicos.....	12
Requisitos operativos.....	13
Requisitos legales.....	14
Requisitos económicos.....	14
1.4. Estudio de alternativas de solución.....	14
Servidor.....	14
Baleike.com.....	15
Joomla + módulos.....	15
Requerimientos.....	15
Drupal + módulos.....	16

Requerimientos.....	16
Wordpress + módulos.....	17
Requerimientos.....	18
Baleike.com/blogak.....	19
1.5. Valoración de las alternativas.....	19
Análisis costes/beneficios del sistema .....	19
Servidor hardware y software.....	19
Baleike.com.....	19
Riesgos en las alternativas del sistema .....	21
Servidor.....	21
Baleike.com.....	21
Paliar riesgos en las alternativas del sistema .....	22
Servidor .....	22
Baleike.com.....	23
1.6. Selección de la solución.....	24
2. Análisis del sistema.....	27
2.1 Definición del sistema.....	27
2.1.1 Requisitos exactos del sistema web .....	27
2.1.2 Entorno tecnológico del sistema .....	28
2.1.3 Normas que cabe seguir en el sistema web .....	29
2.1.4 Identificación de los usuarios del sistema.....	29
2.2. Establecimiento de requisitos.....	30
2.2.1. Definición de los requisitos, casos de uso y asociaciones entre ellos.....	30
2.2.2. Diagramas de los principales casos de uso.....	37
2.3. Definición de interfaces de usuario.....	40
2.3.1. Perfiles de usuario.....	40
2.3.2. Principios generales de la interface de usuario.....	42

2.3.3 Interfaces de usuario de casos de uso.....	43
2.4. Especificación del plan de pruebas.....	46
3. Diseño.....	50
3.1. Arquitectura.....	50
3.1.1. Definición de niveles de arquitectura.....	50
3.1.2. Especificación de estándares, normas de diseño y construcción .....	51
3.1.3. Identificación de subsistemas.....	52
.....	55
3.2 Revisión de casos de uso.....	55
3.2.1 Revisión de los subsistemas según los casos de uso.....	55
3.2.2 Elección de alternativas de componentes y licencias más adecuadas.....	58
3.2.3 Especificaciones de desarrollo y pruebas .....	62
3.2.4. Requisitos de implantación .....	65
4. Desarrollo.....	67
4.1. Planificación de las actividades de desarrollo e integración de sistema .....	67
4.2. Desarrollo.....	68
Instalación y configuración de Wordpress.....	68
Integración de módulos genérica.....	68
Módulos que han requerido modificaciones.....	69
Desarrollos avanzados.....	71
4.3 Documentación.....	85
5. Implantación y mantenimiento.....	85
5.1 Formación.....	86
5.2 Implantación del sistema y pruebas.....	86
5.3 Nivel de servicio.....	87
5.4 Aceptación .....	87
6. Conclusiones.....	88

# 1. Estudio de viabilidad

En el estudio de viabilidad se definirá el alcance del sistema, la situación actual, los objetivos a alcanzar y los requisitos a cumplir. Para luego analizar las alternativas disponibles, analizar las opciones y tomar una decisión.

## 1.1. Establecimiento del alcance del sistema

El proyecto es un trabajo requerido a la empresa prestadora del servicio (en la cual se enmarca mi trabajo) por parte de Baleike. Como antes ya hemos comentado es un medio comunicación local que consta de una página web y una comunidad de blogs. Además en los últimos tiempos han empezado a utilizar una serie de servicios web externos, lo que ha acelerado la necesidad de renovar toda la plataforma.

El objetivo del proyecto por lo tanto sería tras una consultoría y análisis, la actualización, migración e implementación de una plataforma digital que contemple la integración con los diferentes servicios web y la versión móvil de sus sitios.

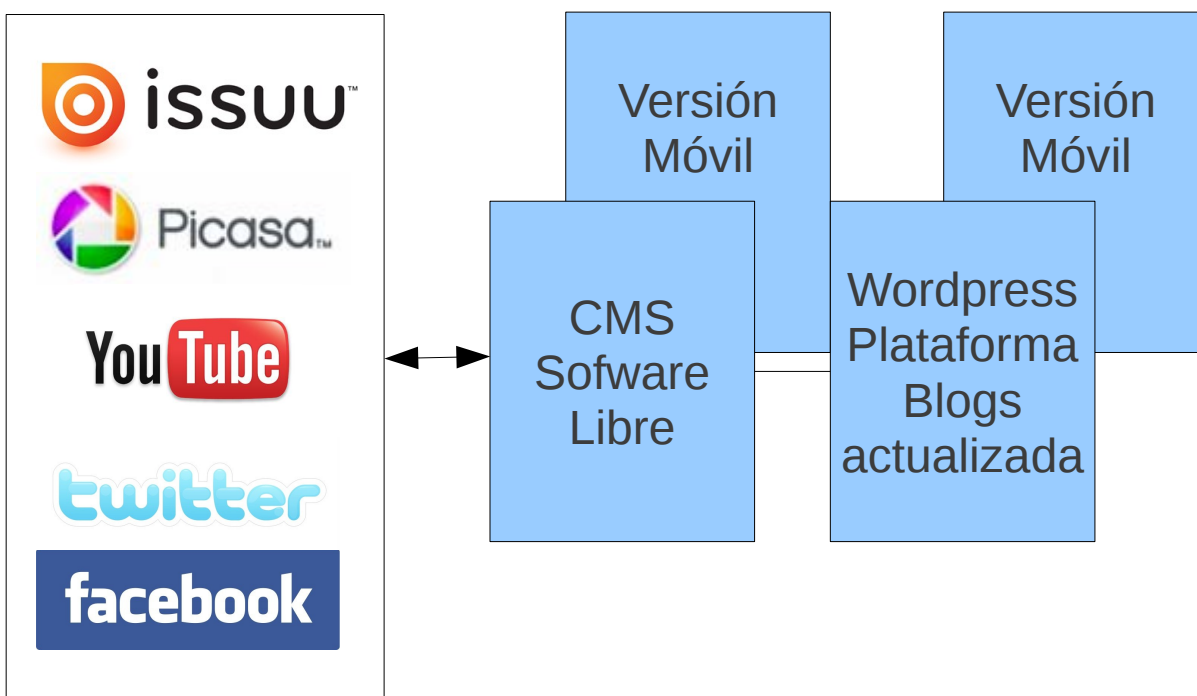
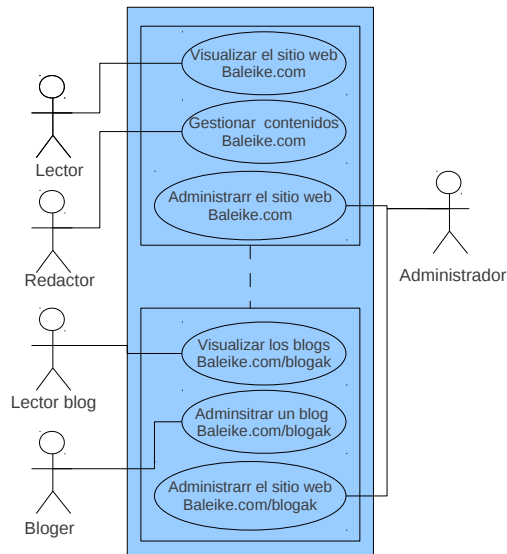


Ilustración 2: Simplificación de los servicios web a integrar en la plataforma

A su vez estos serían los agentes principales que tomarán parte en la plataforma de blogs y el medio de comunicación de una manera simplificada.



*Ilustración 3: Esquema simple de la web principal y la comunidad de blogs*

## **Objetivos detallados**

### **baleike.com**

Este es el sitio principal. Estos deberían de ser sus objetivos desglosados.

- Diseño de la web usando un CMS en software libre con un diseño adaptado al medio de comunicación.
- Versión móvil para el mismo.
- Integración con la galería de Picassa.
- Integración de la galería de YouTube.
- Integración de la galería de Issuu.
- Analizar la integración con Twitter y Facebook en los comentarios y el acceso como usuario al sitio y la implementación de la solución.
- Módulo de agenda, con soporte para el formato iCal.
- Módulo para encuestas.
- Módulo específico “Zumaia sarean” (similar al actual en la web).

## **baleike.com/blogak/**

Estos serán los objetivos de la plataforma de blogs.

- Actualización a la nueva versión de Wordpress de la plataforma de blogs (migración si lo requiriese).
- Creación o re-diseño de una plantilla genérica para todos los blogs y añadir varias plantillas por defecto del CMS Wordpress.
- Versión móvil genérica para los blogs.

Las necesidades operativas obligan a que la solución sea lo más sencilla posible, rápida y sobre todo que no genere demasiada dependencia hacia el proveedor del servicio. Por lo que se anticipa la necesidad de utilizar software libre como base tecnológica para el proyecto.

El uso con anterioridad de servicios web externos a la plataforma que vamos a crear pueden aportar complejidad al proyecto. Por una parte hay que conseguir una cierta integración con los mismos y por otra en cualquier caso habrá que intentar conjugar su uso con la necesidad de mantener una autonomía o integridad para el proyecto. Ya que en algún momento estos servicios pueden llegar a dejar de funcionar. La consultoría que se realizará como parte del proyecto hará hincapié en este aspecto, aunque la última decisión siempre será del cliente.

A nivel técnico vemos que la plataforma anterior está creada sobre un binomio Joomla (web) + Wordpress (blogs). Esto planteará que a nivel técnico se deban analizar diferentes soluciones posibles. Teniendo en cuenta sobre todo las características respectivas a la integración con los servicios web externos, internos y la movilidad.

Por su parte a nivel de hardware también se hará un pequeño análisis de la situación. Sobre todo a problemas sobrevenidos desde antes: lentitud, imposibilidad de manejar ficheros grandes, problemas con envío de correos etc. Dentro de la consultoría se realizará una propuesta que como todas las demás deberá ser ratificada por el cliente.

En los aspectos legales no se vislumbra mayor problemática. El cliente está dispuesto a liberar todo el desarrollo sobre licencias libres. Es importante por su parte que a la hora de desarrollar tomemos en cuenta las diferentes licencias que actuarán. El proyecto no está pensado para ser publicado para ser usado por otros por lo que el problema legal se circunscribe solamente al uso e integración de las diferentes licencias de los módulos, diseños, librerías etc.



En el aspecto económico, el proyecto consta de una dotación de alrededor de 10.000 euros. Nuestra propuesta por lo tanto irá encaminada hacia ese techo. La empresa prestadora de los servicios tomará en cuenta para rebajar parte del coste que el software, modificaciones, módulos etc. que se utilizan se podrán reutilizar en otros proyectos.

En el proyecto tomará parte el equipo de diseño y programación de la empresa. Junto con el equipo de dirección de proyectos. Por lo que deberán de tomarse en cuenta aspectos de viabilidad económica y disponibilidad de esas personas. Sobre todo porque estos equipos estarán trabajando en diferentes proyectos a la vez que lo hacen en este.

El presupuesto inicial resumido y aproximado es:

- Consultoría | 28 horas | 980 euros
- Migración de Baleike, diseño y diseño móvil | 170 horas | 5950 euros
- Migración/actualización de la plataforma de blogs, diseño genérico y diseño genérico móvil | 86 | 3010

Este proyecto ha tomado como base 35 euros/hora. Se ha rebajado 10 euros/hora en el caso del desarrollo y se le ha aplicado la cuota que se utiliza para la consultoría. Ya que se calcula que la empresa reutilizará en otros dos proyectos los desarrollos aquí propuestos. Sobre todo localizaciones, modificaciones etc. que se hagan para Wordpress.

El proyecto por lo tanto se presupuesta en 9940€. Quedan fuera del proyecto y por lo tanto también del presupuesto el hardware a utilizar. Aunque se puedan hacer comentarios y propuestas sobre el mismo. Los gastos en cualquier caso recaerían en el cliente.

## ***1.2. Estudio de la situación actual***

En esta sección se realizará un análisis de la situación actual para poder conocer el estado del proyecto.

### **Identificación de los sistemas actuales**

Los sistemas que se analizarán serán los anteriormente descritos baleike.com (web) y baleike.com/blogak (comunidad de blogs). Para ello tomaremos en cuenta a los actores que participan activamente en su uso. En el caso de la web el director de proyectos relacionados con

Internet del medio de comunicación, así como a una representación de los redactores del mismo. En el caso de la comunidad de blogs, tomaremos en cuenta al director de proyectos relacionados con Internet del propio medio, ya que este se ha encargado de recabar la información de los participantes de la comunidad de blogs.

Se analizará especialmente la web y las posibles alternativas que pueda tener. Sobre todo se tomará en consideración las capacidades de actualización, independencia hacia el proveedor, integración con los servicios web y movilidad. En el caso de la comunidad de blogs no se considerarán alternativas ya que se considera la actual alternativa la mejor. Se analizará la necesidad o no de su actualización para mantener el sistema actualizado.

## Descripción de los sistemas actuales

La plataforma en su conjunto se ejecuta sobre un hosting compartido pero flexible en Dreamhost. Más exactamente es un VPS (Virtual Private Server). Este tipo de hosting permite cambiar las características y recursos de la máquina respecto a las necesidades con facilidad. Sus características actuales son según el proveedor:

Disk Storage	<a href="#">Unlimited TB + 50GB Backups</a>
Monthly Bandwidth	Unlimited TB
RAM	<b>300 MB</b> – 4000 MB
Domains Hosted	Unlimited
Full Shell / SSH / FTP / SFTP	Unlimited
Users	Unlimited
E-Mail Accounts (POP/IMAP/Webmail)	Unlimited
MySQL 5 Databases	Unlimited w/ Shared Hosting Plan, MySQL-optimized VPS, or Dedicated Server
Operating System	Debian Linux

Este tipo de solución es asequible y asegura ciertos recursos para el sistema que en cualquier caso pueden ser ampliados de manera sencilla mediante un simple formulario.

### ***Baleike.com***

Es la web del medio de comunicación. Esta hecha con Joomla 1.5.0 y la suma de unos cuantos

módulos al mismo. Este sistema de código abierto utiliza PHP + MySQL para el sistema. Además de eso HTML + CSS + Javascript para gran parte del diseño.

La web tiene diferentes apartados las noticias, la publicación (posibilidad de descargar la revista en pdf), multimedia (posibilidad de visionar la galería de video propia), comunidad de blogs (enlace a la comunidad de blogs), guía comercial (guía comercial que desaparecerá en el nuevo proyecto), enlace externo a una plataforma de anuncios clasificados.

Por otra parte en los laterales de la web se visualizan mediante el uso de sindicación de contenidos (RSS) varias informaciones de otras webs. De esta manera están hechos los módulos de Zumaia Sarean y Zumartian. Por su parte también aparecen un módulo de encuestas y otro de eventos que no parecen haber sido usados con intensidad.

### ***Balieke.com/blogak***

La plataforma de blogs esta creada con Wordpress y varios plugins añadidos. Esta hecha con WordPress 2.3.1 y la suma de unos cuantos módulos al mismo. Este sistema de código abierto utiliza PHP + MySQL para el sistema. Además de eso HTML + CSS + Javascript para gran parte del diseño.

La página principal tiene diferentes pestañas y funcionalidades. La primera “bloga sortu” da la posibilidad de crear un blog; la segunda “Arauk” describe las normas de uso; la tercera “Laguntza” es la ayuda para el uso de la plataforma; y la última “Blogak” es el listado de blogs.

Por su parte en la portada aparecen las funcionalidades de búsqueda y en un lateral la actividad de la plataforma; los últimos posts y comentarios. En la parte central aparecen los pasos para poder registrarse en la misma.

## **Diagnóstico de los sistemas actuales**

Igual que en los anteriores puntos se diferenciarán la web y la comunidad de blogs.

### ***Baleike.com***

El responsable de los proyectos en Internet del propio medio nos comunica que el *core* del CMS Joomla ha sido tocado. Debido sobre todo a la implantación de la galería multimedia. Una mala decisión por su parte hizo que la plataforma multimedia requiriese de modificaciones que afectaron a la posibilidad de actualizar de manera sencilla el CMS en su conjunto.

Los redactores por su parte nos remarcan la necesidad de que el CMS sea más fácil de utilizar que

el actual. Al parecer se han encontrado diferentes problemas con el editor del mismo. Por su parte consideran de importancia la integración de los diferentes servicios web que utilizan en la plataforma de manera integrada. Especialmente quieren dar la posibilidad de participar en la plataforma con credenciales externas a la plataforma (Twitter, Facebook...) para los comentarios, pero siempre manteniendo a su vez la gestión de usuarios propia.

El diagnóstico general, por lo tanto apunta a la necesidad de actualizar/cambiar el CMS, para lo cual habrá que analizar diferentes alternativas y cumplir con los requerimientos. Sin duda la facilidad de uso, integración con la plataforma, integración los servicios web externos y la movilidad marcarán la elección.

### ***Baleike.com/blogak***

Contrastada la información con el jefe de proyectos de Internet, este nos comunica que ha recogido por parte de los usuarios varias quejas respecto a la usabilidad. Muchas de esas quejas, por no decir todas están subsanadas con el nuevo editor que se incluye en la nueva versión.

Por lo que parece no habrá mayores problemas para actualizar a la nueva versión. Además las únicas quejas que existían sobre el anterior pueden ser subsanadas con la misma actualización. Además de ello, la versión móvil es un requerimiento por lo que como novedad se integrará también la versión móvil seguramente mediante algún *plugin*.

## **1.3. Definición de requisitos del sistema**

Mediante el estudio del sistema de la situación actual de la web y la comunidad de blogs, los puntos de mejora y problemas detectados, y las entrevistas con los usuarios de estos, se han identificado y catalogado los siguientes requisitos (la prioridad de cada uno de ellos está indicada como un número entre 0 y 100, siendo 100 el prioritario).

### **Requisitos técnicos**

(100) Arquitectura: el contenido del sitio web y la comunidad de blogs deberá poderse administrar mediante la utilización de cualquier navegador.

(80) Arquitectura: el contenido del sitio web y la comunidad de blogs deberá estar almacenado en un sistema gestor de bases de datos relacionales, sobre el cual se puedan realizar futuras consultas no previstas en la actualidad. Este sistema de base de datos estará basado en software libre.

(80) Arquitectura: Se deberá integrar a su vez con Twitter y Facebook en los comentarios y el

acceso como usuario al sitio. Teniendo en cuenta la necesidad de mantener también el sistema de gestión de usuarios propio.

(80) Seguridad: el contenido del sitio web única y la comunidad de blogs sólo podrá ser modificado por aquellas personas autorizadas para ello. La plataforma deberá de aportar diferentes roles de seguridadk.

(80) Seguridad: se podrán realizar copias de seguridad por separado y conjuntamente del contenido del sitio web y la comunidad de blogs.

(80) Normativas y/o estándares: el sitio web deberá cumplir con los estándares marcados por el World Wide Web Consortium (HTML, CSS, etc.).

(80) Debido al conocimiento por parte del cliente la solución deberá de utilizar como lenguaje de programación PHP. En los casos que se requiera y en la parte del diseño se podrá utilizar Javascript.

(60) Normativas y/o estándares: el sitio web deberá intentar acercarse en la medida de lo posible a cumplir con las normas de accesibilidad marcadas por el World Wide Web Consortium (Web Accessibility Initiative). Manteniendo un equilibrio.

## **Requisitos operativos**

(100) Operativa: el sitio web y la comunidad de blogs deberá ser visualmente atractivo.

(100) Operativa: el sitio web y los blogs de la comunidad de blogs deberán poder ser visualizados en dispositivos móviles, en una versión reducida de la web.

(100) Operativa: la gestión del contenido del sitio web y los blogs de la comunidad de blogs deberá poder ser realizada por una persona no técnica, es decir, que no tenga conocimientos de HTML, JavaScript, etc., de forma fácil e intuitiva.

(90) Arquitectura: el sitio web tendrá que ser capaz de interactuar con servicios web externos como Picassa, YouTube, Issuu... Para mostrar

(90) Operativa: el sitio web deberá tener una estructura clara, dando importancia a la portada sobre la sobre-estructuración del contenido.

(90) Operativa: la comunidad de blogs deberá mantener sus funcionalidades actuales y la estructura actual.

(90) Operativo: tanto el gestor de contenidos, como la comunidad de blogs deberán de poder ser localizados al euskara. En el caso de que no existan las localizaciones necesarias el sistema debe de

poder añadirlas, para lo cual deberá de estar preparado.

(80) Administración: la administración del sitio web y la comunidad de blogs deberá poder realizarse a través de un navegador web.

## **Requisitos legales**

(60) La licencia de uso del software de gestión de contenidos debe ser lo menos restrictiva posible.

(60) La licencia de uso de la base de datos debe ser lo menos restrictiva posible.

(60) Los manuales, código y documentación generados deberán de ser lo menos restrictivos posibles.

(60) La licencia de uso del sistema operativo del servidor web debe ser lo menos restrictiva posible.

## **Requisitos económicos**

(100) El proyecto en su conjunto deberá adaptarse al presupuesto presentado.

(90) Las partes centrales del proyecto como son el CMS y el software para la comunidad de blogs deberán estar exentos de costes por licencia de uso.

(90) Las partes centrales en lo que respecta a la arquitectura como son el sistema de bases de datos, el servidor web y el sistema operativo del mismo deberán de estar exentos de costes por licencia de uso.

(80) En el caso de ser necesario un gasto en concepto de licencia de uso de alguno de los componentes menores del sistema, éste deberá ser lo más pequeño posible y deberá de tomar en consideración otros requisitos antes de tomar la decisión final.

## **1.4. Estudio de alternativas de solución**

### **Servidor**

El servidor que se propone como alternativa al VPS que se utiliza en la actualidad, podría ser un servidor pequeño dedicado que cubra las necesidades del sistema.

Las características del mismo deberían de igualar las actuales del VPS y tomar en cuenta los

requisitos que se esperan a nivel de software. En cualquier caso el servidor correría sobre un sistema operativo GNU/Linux y un servidor Apache. Además seguramente tendría instalada como base de datos MySQL y estaría preparado para usar el lenguaje PHP con todas las librerías necesarias para ello.

Una propuesta podría ser la de utilizar un pequeño servidor dedicado que podría tener por ejemplo estas características:

<b>Marca</b>	Intel
<b>Modelo</b>	Celeron / Atom
<b>Frecuencia</b>	1.20+ GHz
<b>Arquitectura</b>	64 bits
<b>NIC</b>	FastEthernet
<b>RAM</b>	2 GB

## **Baleike.com**

El sitio web principal nos obliga a analizar diferentes opciones. Para ello y tomando en cuenta los requerimientos antes expuestos haremos una pre-selección de gestores de contenidos, todos ellos utilizando PHP como base para la programación. Las alternativas propuestas pueden ser tres. Son de los tres CMS más utilizados, Joomla, Drupal y Wordpress.

### ***Joomla + módulos***

Es un gestor de contenidos muy utilizado para la creación de portales web. Su página web es <http://joomla.org>. La última versión del CMS es la 1.7. El sistema además incluye la posibilidad de utilizar muchos de los desarrollos de otros integrándolos mediante añadidos. De esta manera se podría llegar a cumplir con todos o casi todos los requerimientos operativos. Estos son en resumen los requerimientos y su estado con esta alternativa:

## **Requerimientos**

### **Técnicos**

- PHP 5.2.4+

- MySQL 5.0.4+
- Apache (mod\_mysql, mod\_xml, mod\_zlib) 2.x+

## **Operativos**

Cumple con la mayoría de requisitos especificados. Algunas funcionalidades habría que integrarlas utilizando extensiones y módulos complementarios. La mayoría existen aunque no se descarta la necesidad de tener que pagar alguno de los mismos.

Por otra parte en su contra existen los antecedentes de dificultades de uso por parte de algunos usuarios de la plataforma. Además de las reticencias a seguir utilizando el sistema aunque este sea bueno. No hay demasiadas mejoras en lo que respecta al editor y la gestión.

## **Legales**

Utiliza una licencia GNU/GPL <http://www.gnu.org/licenses/gpl-2.0.html> por lo que no se anticipan problemas en este sentido.

Los módulos utilizados además son considerados obras derivadas, por lo que no habría problema para integrarlos en nuestros desarrollos.

## **Económicos**

El coste de disponibilidad es cero ya que se puede descargar desde la página web.

Algunos módulos utilizados podrían suponer un coste añadido, aunque bajo.

El coste de la migración no parece que sea muy grande, ya que es otra versión del mismo software.

## ***Drupal + módulos***

Es otro gestor de contenidos muy utilizado para la creación de portales web. Su página web es <http://drupal.org>. La última versión del CMS es la 7.9. El sistema además incluye la posibilidad de utilizar muchos de los desarrollos de otros integrándolos mediante añadidos. De esta manera se podría a llegar a cumplir con todos o casi todos los requerimientos operativos. Estos son en resumen los requerimientos y su estado con esta alternativa:

## **Requerimientos**

### **Técnicos**

- PHP 5.2.5+



- MySQL 5.0.15+
- Apache 1.3+

## **Operativos**

Cumple con la mayoría de requisitos especificados. Algunas funcionalidades habría que integrarlas utilizando módulos complementarios. La mayoría existen aunque no se descarta la necesidad de tener que pagar y/o desarrollar alguno de los mismos.

La accesibilidad y cumplimiento de algunos estándares parece ser el fuerte de la herramienta. Por otra parte su administración no es del todo intuitiva para personas con pocos conocimientos técnicos.

## **Legales**

Utiliza una licencia GNU/GPL <http://www.gnu.org/copyleft/gpl.html> por lo que no se anticipan problemas en este sentido.

Los módulos utilizados además son considerados obras derivadas, por lo que no habría problema para integrarlos en nuestros desarrollos.

## **Económicos**

El coste de disponibilidad es cero ya que se puede descargar desde la página web.

Algunos módulos utilizados podrían suponer un coste añadido, aunque bajo.

El coste de la migración podría ser bastante grande ya que requeriría una modificación total e integración en la nueva base tecnológica de la base de datos.

## **Wordpress + módulos**

Es un gestor de contenidos utilizado hasta ahora para la creación de blogs. Aunque desde su versión 3.0 ha cambiado mucho hasta convertirlo en más intuitivo, integral y *multisite*. Lo que hace a esta herramienta muy adecuada para crear blogs, pero también página y portales de tamaño medio. Su página web es <http://wordpress.org>. La última versión del CMS es la 3.2.1. El sistema además incluye la posibilidad de utilizar muchos de los desarrollos de otros integrándolos mediante añadidos. De esta manera se podría llegar a cumplir con todos o casi todos los requerimientos operativos. Estos son en resumen los requerimientos y su estado con esta alternativa:

## Requerimientos

### Técnicos

- PHP 4.3+
- MySQL 4.1.2+
- Apache

### Operativos

Cumple con la mayoría de requisitos especificados. Algunas funcionalidades habría que integrarlas utilizando módulos complementarios. La mayoría existen aunque no se descarta la necesidad de tener que pagar y/o desarrollar alguno de los mismos.

Da gran posibilidad y autonomía de desarrollo. Muchas de las funcionalidades ya están implementadas mediante módulos. Algunas incluso tienen varias opciones que habría que analizar también en profundidad. Una clave sería la integración para acceder como usuario en el sistema con credenciales externas utilizando también el sistema de usuarios actual. Lo trataremos más a fondo en el siguiente punto si está es la solución escogida. La administración y gestión es muy intuitiva y editor es sumamente sencillo. Parece muy adecuado para los fines propuestos para un medio de comunicación digital de tamaño medio.

### Legales

Utiliza una licencia GNU/GPL <http://wordpress.org/about/gpl/> por lo que no se anticipan problemas en este sentido.

Los módulos utilizados además son considerados obras derivadas, por lo que no habría problema para integrarlos en nuestros desarrollos.

### Económicos

El coste de disponibilidad es cero ya que se puede descargar desde la página web.

Algunos módulos utilizados podrían suponer un coste añadido, aunque bajo.

El coste de la migración podría ser grande aunque hay herramientas como <http://downloads.wordpress.org/plugin/joomla-to-wordpress-migrator.1.5.4.zip> que lo facilitan enormemente. Habría que tomar en cuenta que esta es la misma herramienta que utiliza la comunidad de blogs por lo que podría ser interesante la integración de toda la plataforma en una misma herramienta.

## **Baleike.com/blogak**

No se proponen alternativas a la comunidad de blogs, ya que se considera que Wordpress es en la actualidad la herramienta de software libre más avanzada para ello. Se propone sin embargo la actualización a la nueva versión de la misma. Las características, así como sus requerimientos están explicados en el anterior punto.

### **1.5. Valoración de las alternativas**

En este apartado se analizarán las diferentes alternativas, tanto desde el punto de vista económico como de los riesgos asociados.

## **Análisis costes/beneficios del sistema**

### ***Servidor hardware y software***

Se compararán la solución actual que utiliza un hosting VPS, con la propuesta de utilizar un servidor dedicado barato. Se tomarán en cuenta los gastos de administración y seguridad, gastos del dominio y los costos asociados propios del servidor.

Hosting VPS:  $0^* + 0^{**} + 15\$/mes = 133€/año$

Servidor dedicado:  $400 € + 9€/año + 15€/mes = 589€/año$

En el hosting VPS se considera que no hay gastos de administración y seguridad esceptuando si hubiera problemas, para lo cual se estipularía un precio. Pero digamos que las medidas de seguridad básicas están implantadas gracias al paraguas que ofrece el hosting que se engarga de ellas. Por su parte el dominio es gratuito en este hosting. El precio final por lo tanto en un año sería de 133 €/año

En el servidor dedicado se considera que se necesitarían 4 horas \* 50 €/hora para implantar el sistema y otras 4 horas \* 50 €/hora para implantar las medidas de seguridad y mejoras del sistema. 8 horas por lo tanto que ascienden a 400 €. A lo que hay que sumarle el dominio y el coste de la máquina. En total 589 € el primer año por lo menos.

### ***Baleike.com***

Se compararán los costos de las tres propuestas. Se tomarán en cuenta sobre todo los costes de implantación, costes de migración y coste de adaptación. El coste de implantación hace referencia a la instalación y configuración del sistema. El coste de migración a la facilidad o dificultad asociada

al proceso de migración. El coste de adaptación las facilidad o dificultad de utilizando código asociado al producto conseguir los requerimientos expuestos. Estas valoraciones se harán sujetas a los datos remarcados en los puntos anteriores y después de haber hecho un análisis de las diferentes alternativas del sistema.

Quedan fuera de este punto los costes de disponibilidad ya que en todo los casos son cero. Debido a que se utilizará, porque así se ha requerido sólo soluciones basadas en software libre para el proyecto.

Joomla = 140€ + 280€ + 2800€ = 3220€

Drupal = 140€ + 1120€ + 4200€ = 5460€

Wordpress = 140€ + 560€ + 2800€ = 3500€

En el primer paso se considera que se utilizarán 4 horas a 35€/hora para la instalación y configuración mínimas del sistema. Por su parte se considera que el coste de migración será sencillo ya que el cambio es simplemente de versión. En cambio el coste de adaptación es bastante grande, debido a la complejidad del CMS, la necesidad de analizar los modulos añadir y sobre todo porque no se puede reutilizar gran parte del código creado con anterioridad. Esto es así porque se modificó el *core* del sistema. Se presupone la necesidad de utilizar por los menos 80 horas de trabajo en esta adaptación. Por lo tanto el precio final es de 3220€.

En el segundo caso el coste de implantación es el mismo, pero el coste de migración es muy superior. Se considera que se pueden utilizar hasta 32 horas para la migración de la plataforma al nuevo CMS. A su vez el coste de adaptación también es mayor, debido a los problemas que pueden sobrevenir de la migración además de la complejidad de Drupal. Comparando con los otros no existen tantos módulos añadidos, por lo que alguna de las funcionalidades podría tener que ser desarrollada por nosotros.

El últimos caso el coste de implantación es el mismo, pero el coste de migración es más costoso que en el primero, aunque no tanto como en el segundo. Se considera que Wordpress y las herramientas que antes hemos descrito podrían facilitar la migración bastante, por lo que se le asignan 16 horas que a 35€/hora hacen 560€. Por su parte la adaptación es parecida a la del primer ejemplo ya que Wordpress cuenta con un montón de *plugins* que podrían ahorrarnos en desarrollar gran parte de las funcionalidades requeridas. Todo ello hace que la solución basada en Wordpress ascienda a 3500€.

## Riesgos en las alternativas del sistema

### *Servidor*

Estos son los riesgos asociados a la diferentes propuestas de hardware:

#### **Hosting VPS**

- Capacidad de hardware reducida, debido a la poca RAM que se reserva para el hosting.
- Inseguridades asociadas por el mal uso de las aplicaciones o ataques contra las aplicaciones ya instaladas en el sistema.
- Posible desaparición de la empresa proveedora.

#### **Servidor dedicado**

- Fallos de seguridad devenidos de las instalación del sistema operativo y de las medidas de seguridad en el mismo.
- Inseguridades asociadas por el mal uso de las aplicaciones o ataques contra las aplicaciones ya instaladas en el sistema.
- Posible desaparición de la empresa proveedora.

### *Baleike.com*

Estos son los riesgos asociados a la diferentes propuestas de software:

#### **Joomla**

- Dificultades **menores** no contempladas en la migración de software.
- No disponibilidad de módulos para las funcionalidades requeridas o coste excesivo de las mismas.
- Desaparición del equipo de desarrolladores principal de la plataforma.

#### **Drupal**

- Dificultades **añadidas** y no contempladas en la migración de software.
- No disponibilidad de módulos para las funcionalidades requeridas o coste excesivo de las mismas.
- Desaparición del equipo de desarrolladores principal de la plataforma.

## **Wordpress**

- Dificultades **posibles** no contempladas en la migración de software.
- No disponibilidad de módulos para las funcionalidades requeridas o coste excesivo de las mismas.
- Desaparición del equipo de desarrolladores principal de la plataforma.

## **Paliar riesgos en las alternativas del sistema**

### ***Servidor***

Estos son las soluciones a los riesgos asociados a la diferentes propuestas de hardware:

### **Hosting VPS**

- El propio hosting prevé la posibilidad de ampliar la RAM y otros recursos simplemente con un formulario. Por lo que el problema se convierte simplemente en económico y los precios son razonables. Por su parte el incremento puede ser ajustado y escalado según la necesidad sin tener que migrar el sistema.
- Se ofrecerá una cuota de mantenimiento por horas o por bonos al cliente para subsanar los posibles problemas de seguridad que pudieran crearse en las aplicaciones instaladas. Esto queda fuera del presupuesto, menos en el periodo de prueba. La cuota que se marcará por hora será de 50€/hora.
- Disponibilidad de un sistema de Backup automático por si la empresa desapareciera. El sistema en si ya está preparado con un sistema de Backup, pero convendría tener uno extra fuera del proveedor por si este desapareciera de manera inesperada. Esto tendría un coste asociado igual o parecido al de tener otro hosting VPS o servidor dedicado. Dejaríamos en manos del cliente elegir la opción.

### **Servidor dedicado**

Algunas de los riesgos son los mismos que en el anterior, los que no lo son están explicados aquí:

- La implantación del sistema operativo y las medidas de seguridad asociadas en un servidor puede requerir la monitorización y mantenimiento del sistema. No ya en el plano de las soluciones instaladas, sino en el propio sistema. Para ello se ofrecerá una cuota de mantenimiento por horas al cliente para asegurarnos del buen funcionamiento del sistema

operativo y las medidas de seguridad asociadas. Esto queda fuera del presupuesto, menos en el periodo de prueba. La cuota que se marcará por hora será de 50€/hora.

## **Baleike.com**

Estos son las soluciones a los riesgos asociados a la diferentes propuestas de software. Debido a que todas las soluciones tienen problemas parecidos se harán matices en los que los tengan.

- Las dificultades en la migración son intrínsecas a cada plataforma:
  - Joomla: fácil
  - Drupal: difícil
  - Wordpress: intermedio

Teniendo esto en cuenta lo único que subsanará más dificultades asociadas será un buen análisis de la plataforma de antemano. En el caso de problemas mayores deberíamos por nuestra parte tener contacto con equipos de desarrolladores externos con experiencia en este tipo de migraciones.

- La no disponibilidad de módulos es un riesgo a tomar en cuenta en dos vertientes, la facilidad o dificultad de encontrarlas y/o de desarrollarlas. La primera es una cuestión de análisis mientras la segunda queda en parte en nuestras capacidades y en la de las de la plataforma. Se considera que estos son las características de cada una:
  - Joomla: fácil | -fácil
  - Drupal: -fácil | fácil
  - Wordpress: +fácil | +fácil

Teniendo esto en consideración deberíamos de tener en cuenta que plataforma escoger bien y para eso hacer un esfuerzo en el análisis. Que por otra parte ya esta incluido en el proyecto a modo de consultoría. En el caso de todas formas de necesitar algún desarrollo, deberíamos de analizar según la solución que adoptemos si necesitamos alguien que nos apoye. Para lo cual estaría bien tener en cuenta desarrolladores externos que pudieran ayudarnos.

- Desaparición del equipo de desarrolladores principal de la plataforma es un problema asociado a la mayoría del software libre. Aún así muchos de estos proyectos además de comunidad suelen tener empresas asociadas que les dan más apoyo si cabe. Para solucionar este problema deberíamos afinar en el análisis y la selección de la solución y en cualquier

caso asegurarnos que el sistema y la documentación asociada son suficientemente buenos como para en caso de desaparición poder actuar. A su vez podemos plantearnos la necesidad de apoyo externo en el caso que algo así ocurriera.

## **1.6. Selección de la solución**

En la toma de la decisión hemos tenido en cuenta todo lo antes expuesto. La descripción general y alcance del proyecto, la situación actual del sistema, los requisitos que deberá cumplir la solución adoptada, la descripción de las soluciones alternativas consideradas y el análisis de costes/beneficios de las diferentes soluciones y riesgos asociados a cada una de ellas.

En lo que se refiere al hardware se escoge seguir en la situaciónpruebas actual. El hecho de que en el análisis de coste/beneficios haya sido desfavorable para la solución de servidor dedicado junto con las posibilidades de escalar del hosting VPS, hacen de este de momento la mejor opción. Sin duda el servidor dedicado podrá ser una solución a futuro, en el caso que las necesidades así lo requieran.

Respecto a los requisitos presentados todas las soluciones cumplen más o menos con los mismos. Algunos como Joomla y Drupal flaquean un poco en la facilidad de uso al nivel de usuario pero también de administración del gestor. Joomla además tiene reticencias por parte de usuarios actuales de la plataforma.

Por su parte son dos los aspectos más importantes. La integración que se pueda conseguir con los servicios web externos, pero también entre los sitios de la plataforma (sitio web y comunidad de blogs), para lo que es importante alta disponibilidad de módulos disponibles. Y por otra parte la facilidad o no de migración de la plataforma. En lo primero y segundo es claramente ganador Wordpress ya que aporta una amplia comunidad de apoyo. Joomla también tiene su punto fuerte aquí ya que sería mucho más fácil actualizar, aunque sea empezando casi de cero una web que haciendolo de otra manera.

Si hacemos referencia a los costes/beneficios vemos que la solución más económica en principio es la de actualizar Joomla por su facilidad para la migración sobre todo. En este caso el presupuesto para Drupal, está casi fuera de lo posible, mientras que la solución basada en Wordpress se sitúa un poco por encima de Joomla pero podría ser asumible.

En el apartado de riesgos aparte de los ya citados con anterioridad (aunque son más ventajas que riesgos) todos más o menos rivalizan. En este caso la mayor importancia por lo tanto la adquiere la



necesidad de asegurar el futuro de la plataforma. Para lo cual es necesario medir la capacidad de la comunidad y de las empresas asociadas para avanzar en un futuro con la plataforma. De esta forma nos aseguraríamos el cumplimiento a futuro de muchos de los requisitos que puedan aparecer, pero sobre todo reduciríamos de manera drástica los riesgos.

Para ello hemos consultado la web ohloh.net. En esta web se recogen y comparan diferentes métricas de diferentes proyectos de software libre. En esta web hemos comparado nuestras tres propuestas para comparar sobre todo su estado actual pero sobre todo su posible desarrollo futuro. Aquí se puede consultar la comparativa [http://www.ohloh.net/p/compare?project\\_0=WordPress&project\\_1=Joomla%21&project\\_2=Drupal+%28core%29](http://www.ohloh.net/p/compare?project_0=WordPress&project_1=Joomla%21&project_2=Drupal+%28core%29).

Compare Projects			
	WordPress <small>Remove this project</small>	Joomla! <small>Remove this project</small>	Drupal (core) <small>Remove this project</small>
<b>General</b>			
Ohloh Data Quality	✔ Updated about 6 hours ago	⚠ Updated about 1 month ago	⚠ Updated 19 days ago
Homepage	<a href="http://wordpress.org">wordpress.org</a>	<a href="http://www.joomla.org">www.joomla.org</a>	<a href="http://drupal.org">drupal.org</a>
Project License	GPL	GPL 2+	GPL
Estimated Cost	\$2,101,580.00	\$13,409,209.00	\$8,572,496.00
<b>All Time Activity</b>			
Committers (All Time) <small><a href="#">View as graph</a></small>	<a href="#">41 developers</a>	<a href="#">66 developers</a>	<a href="#">45 developers</a>
Commits (All Time) <small><a href="#">View as graph</a></small>	<a href="#">17,554 commits</a>	<a href="#">20,933 commits</a>	<a href="#">74,427 commits</a>
Initial Commit	✔ over 8 years ago	✔ about 6 years ago	✔ over 11 years ago
Most Recent Commit	✔ about 14 hours ago	✔ about 1 month ago	✔ 20 days ago
<b>12 Month Activity</b>			
Committers (Past 12 Months)	✔ <a href="#">16 developers</a>	✔ <a href="#">10 developers</a>	✔ <a href="#">10 developers</a>
Year-Over-Year Commits	✔ Stable	⚠ Decreasing	⚠ Decreasing
<b>30 Day Activity</b>			
Committers (Past 30 Days)	📄 <a href="#">9 committers</a>	📄 <a href="#">5 committers</a>	📄 <a href="#">5 committers</a>
Commits (Past 30 Days)	<a href="#">318 commits</a>	<a href="#">102 commits</a>	<a href="#">225 commits</a>
Files Modified	213 files	402 files	337 files
Lines Added	5,413 lines	3,354 lines	14,412 lines
Lines Removed	4,118 lines	2,246 lines	5,488 lines

Ilustración 4: Comparativa de soluciones propuestas

En ella podemos ver que todos los proyectos gozan de bastante salud, aunque se ve claramente la progresión de Wordpress frente a otros. Los números de los últimos tiempos, el número de desarrolladores en la actualidad y las actualizaciones hacen ver que Wordpress es una alternativa a futuro. El indicador más claro a este respecto es el que marcan los envíos al repositorio de cada año donde Wordpress aparece estable, frente a Joomla y Drupal que decrecen.

Por lo tanto y como conclusión la solución que se propone es la de Wordpress. En la parte de requisitos cumple con la mayoría y sino con la gran gama de modulos que hay para la comunidad la mayoría se podrían realizar. No habría problema y de hecho está estimado hacer modificaciones o

desarrollos propios.

A nivel de requisitos sobre todo tienden la balanza hacia aquí la integración con los servicios web externos (gracias a los plugins antes citados) pero también la posibilidad que daría de integrar los dos sitios que componen la plataforma en uno. De esta manera la gestión de la plataforma sería integral ya que tanto el sitio web como la comunidad de blogs usarían la misma tecnología.

El análisis de coste/beneficio aunque no es el más barato es asumible por el presupuesto. Merece la pena apostar por una solución integral ya que esto también beneficiará económicamente a la hora de mantenimiento, reutilización, administración....

A su vez los riesgos que se asumen en la actualidad son bastante parecidos en todos los casos y a futuro desde luego Wordpress parece la mejor solución una vez vistos los datos. Una vez vistos todos estos datos por lo tanto la decisión sería utilizar Wordpress como base tecnológica para nuestro proyecto.

## **2. Análisis del sistema**

En el análisis del sistema describiremos en mayor profundidad la plataforma Baleike.com. En el anterior punto se decidió utilizar Wordpress como base tecnológica para la plataforma, aquí se describirá la misma en más profundidad.

En el anterior punto se trató específicamente las dos partes de la plataforma, ya que utilizaban diferentes tecnologías. En esta ocasión y aunque se harán comentarios diferenciados en algunos casos, se tomará la plataforma en su integridad, ya que la solución propuesta lo hace posible.

### **2.1 Definición del sistema**

En este apartado, se describirá el sistema, establecerá cómo se comunicará con otros en caso de ser necesario y qué usuarios serán representativos en el uso del mismo.

#### **2.1.1 Requisitos exactos del sistema web**

Estos son los requisitos que deberá de cumplir el sistema.

- La plataforma (sitio web principal y la comunidad de blogs) deberá poderse administrar mediante los navegadores Mozilla Firefox (versión 3.6+), Internet Explorer (versión 7.0+), Google Chrome, Safari y Opera.
- La plataforma deberá ser visualmente atractiva. El sitio web principal deberá tener una estructura clara, dando importancia a la portada sobre la sobre-estructuración del contenido. La comunidad de blogs deberá mantener sus funcionalidades actuales y la estructura actual.
- El contenido de la plataforma sólo podrá ser modificado por aquellas personas autorizadas para ello. La plataforma deberá de aportar diferentes roles que serán especificados más adelante en la sección “Identificación en la plataforma”.
- La plataforma podrá ser visualizada en dispositivos móviles, en una versión reducida de la web.
- La gestión del contenido de la plataforma deberá poder ser realizada por una persona no técnica, es decir, que no tenga conocimientos de HTML, JavaScript, etc., de forma fácil e intuitiva. Para lo que será necesario un editor WYSIWYG (What You See Is What You Get).
- La administración de la plataforma deberá poder realizarse a través de un navegador web y

de manera intuitiva.

- El acceso a la misma se deberá integrar a su vez con Twitter y Facebook en los comentarios y el acceso como usuario al sitio. Teniendo en cuenta la necesidad de mantener también el sistema de gestión de usuarios propio.
- La web principal de la plataforma tendrá que ser capaz de interactuar con servicios web externos como Picassa, YouTube, Issuu...
- La web principal de la plataforma tendrá diferentes páginas que serán accesibles mediante menús. Cada una de estas páginas será personalizable y editable desde la gestión.
- Creación de encuestas y posibilidad de añadirla a la página principal.
- Creación de eventos posibilidad de añadirla a la página principal.
- Creación de blogs dentro de la plataforma.
- La plataforma estará localizada al euskara.
- La licencia de uso del software de gestión de contenidos sera GNU/GPL o compatible.
- La licencia de uso de la base de datos debe ser GNU/GPL o compatible.
- La licencia de uso del sistema operativo del servidor web debe ser GNU/GPL o compatible.
- Los manuales, código y documentación generados deberán de ser lo menos restrictivas posibles.

### **2.1.2 Entorno tecnológico del sistema**

El entorno tecnológico del sistema web será el siguiente:

- El sistema operativo será GNU/Linux (distribución a definir entre posibles Debian, Fedora o Ubuntu server)
- La plataforma tendrá como base tecnológica el gestor de contenidos Wordpress 3.2+.
- El contenido de la plataforma deberá estar almacenado en un sistema gestor de bases de datos relacional MySQL (versión 4.1.2+).
- La plataforma deberá cumplir con los estándares marcados por el World Wide Web Consortium (HTML, CSS 2.0, etc.) actuales. Tomando especial consideración la evolución de los mismos a futuro (la futura transición a HTML5 y CSS3; asumiendo que el coste no

deberá ser demasiado grande).

- El servidor web que se utilizará en primera instancia será Apache.
- La plataforma deberá de utilizar como lenguaje de programación PHP (versión 4.3+) en los desarrollos. En los casos que se requiera y en la parte del diseño se podrá utilizar Javascript.

### 2.1.3 Normas que cabe seguir en el sistema web

Las normas y estándares que hay que seguir en la implementación del sistema web serán las siguientes:

- En cuanto al sistema operativo, se seguirá el proceso habitual de “Instalación de servidores GNU/Linux” de Elurnet S.L.
- La plataforma no deberá pasar a producción sin antes haberse probado en los servidores de desarrollo. Más información en la “Especificación del plan de pruebas”.
- Todos los desarrollos estarán condicionados a la integración con Wordpress. Por lo que utilizarán tecnologías PHP, HTML, CSS y Javascript. Será indispensable seguir sus recomendaciones y métodos de programación. Las recomendaciones ha seguir son las siguientes [http://codex.wordpress.org/Main\\_Page](http://codex.wordpress.org/Main_Page)
- Los añadidos a la plataforma se realizarán a poder ser en forma de *Plugins* de Wordpress, para evitar así interferir en la base de la plataforma. Se tomará en cuenta la documentación y recomendaciones de la plataforma <http://codex.wordpress.org/Plugins>.
- Las plantillas de diseño que se manejarán seguirán a su vez la documentación y recomendación de la plataforma. Especialmente el uso de *child-themes*. Plantillas que extienden otras plantillas que pueden ser actualizada sin modificar su estructura principal. Estas son las recomendaciones a seguir en este caso [http://codex.wordpress.org/Theme\\_Development](http://codex.wordpress.org/Theme_Development)
- Los desarrollos e implantaciones que queden fuera de estas normas deberán ser documentados mediante los procesos habituales de Elurnet S.L.

### 2.1.4 Identificación de los usuarios del sistema

Estos son los grupos de personas que interactúan con la plataforma.

- Administrador/es de la plataforma: El encargado de la gestión y administración de la

plataforma y su buen funcionamiento. Tiene ciertos conocimientos técnicos sobre la plataforma y conocimientos de gestión de la plataforma de manera integral.

- Redactor/es en la página principal: Son aquellas personas que pueden modificar el contenido y algunos aspectos de la plataforma, aunque no los más sensibles en la página principal. Suelen ser periodistas y opcionalmente sus trabajos pueden ser corregidos o auditados por otros redactores o por el administrador.
- Administrador/es de blog: Puede modificar el contenido y algunos aspectos de la plataforma, sólo en su blog.
- Colaboradores: Son aquellas personas que pueden hacer aportaciones de contenido, aunque éstas deben ser aprobadas.
- Usuarios: Son lectores que han participado alguna vez en el sistema, por ejemplo haciendo comentarios. Deben de estar identificados en la plataforma.

La decisión será tomada sobre todo por los dos primeros grupos, con ayuda de las recomendaciones que puedan hacer los administradores de blog y los colaboradores. La aportación de los usuarios se tomarán en cuenta a posteriori en base a encuestas y recogida de opiniones.

## **2.2. Establecimiento de requisitos**

En el establecimiento de requisitos se especifican dos subelementos que conforman la plataforma. La web principal y la comunidad de blogs. La web principal no es nada más que un blog, pero que tiene en su diseño y visualización especificidades. En los casos particulares que se requiera se hablara de web principal y no de la plataforma en su conjunto.

### **2.2.1. Definición de los requisitos, casos de uso y asociaciones entre ellos**

- El contenido de la plataforma (sitio web principal y la comunidad de blogs) deberá poderse administrar mediante los navegadores Mozilla Firefox (versión 3.6+), Internet Explorer (versión 7.0+), Google Chrome, Safari y Opera.
  - Definición: Se contempla la necesidad de que la herramienta pueda ser utilizada con los navegadores más modernos. Se descarta la compatibilidad hacia atrás con Internet Explorer 6.0 debido a su costes, su poca implantación y sobre todo por las dificultades añadidas al diseño. Se prefiere orientar la herramienta por lo tanto hacia los nuevos

navegadores por parte de los administradores.

- Casos de uso: Visualización de la plataforma

El usuario visualiza la plataforma desde un navegador.

- Asociaciones: todo.

- La plataforma deberá ser visualmente atractiva. El sitio web principal deberá tener una estructura clara, dando importancia a la portada sobre la sobre-estructuración del contenido.

- Definición: Se decide por parte de la administración junto con las aportaciones de la redacción la necesidad de una portada que contenga mucha información. Se propone la necesidad de un encabezado con opciones de búsqueda, publicidad y envío de información, que incluya la posibilidad de destacar una noticia. Una parte central en la izquierda y dos columnas editables a la derecha con diferentes contenidos. En la primera columna se proponen contenidos destacados (noticias, videos, imágenes, encuesta). En la segunda y última columna en cambio se propone integrar un formulario, el bloque zumartian, la agenda,. Aunque todos estos bloques deben ser posibles de mover y modificar su orden y columna.

- Casos de uso: Visualización de la plataforma

Asociaciones: Integración de servicios web externos.

- La plataforma podrá ser visualizada en dispositivos móviles, en una versión reducida de la web.

- Definición: Los redactores y administrador nos proponen el salto de la web a los dispositivos móviles. Para ello se preparará una plantilla específica que cargará sólo cuando los usuarios se conecten desde estos dispositivos. La misma no cargará toda la información, sino solamente las informaciones que aparecen en la columna principal de contenido y la búsqueda. Opcionalmente se podrán visualizar otro tipo de contenidos que aparecen en los menús.

- Casos de uso: Visualización de la plataforma mediante dispositivos móviles.

El usuario lector accede mediante un dispositivo móvil y visualiza la plataforma desde el mismo en una versión reducida de la misma.

- Asociaciones: Integración de servicios web externos.

- La plataforma estará localizada al euskara.
  - Definición: La plataforma íntegramente en su parte de administración y visualización debe estar localizada al euskara. Se acuerda que la parte de administración no es totalmente prioritaria, aunque ya existe localización oficial (aunque no del todo completa de la misma). Sobre todo la parte de visualización de la plataforma es clave.
  - Casos de uso: no aplica.
  - Asociaciones: todo
- La web principal de la plataforma tendrá diferentes páginas que serán accesibles mediante menús. Cada una de estas páginas será personalizable y editable desde la gestión.
  - Definición: Se acuerda que serán:
    - Argitalpenak (publicaciones): Se mostrarán las publicaciones pasadas, que están subidas a Issu.
    - Multimedia: Se mostrará la galería de videos subidos a YouTube
    - Blogs: Se enlazará con la plataforma de blogs.
    - Baleike: Página con información sobre la publicación.
  - Casos de uso: Visualización de la plataforma
  - Asociaciones: Visualización de la plataforma.
- La página web principal tendrá que ser capaz de interactuar servicios web externos como Picassa, YouTube, Issuu...
  - Definición: La plataforma deberá de poder visualizar imagenes de Picassa, video de YouTube, documentos de Issuu y un formulario de Google Groups. Se podrá hacer de manera nativa integrando los códigos *embed* que se aportan en las mismas plataformas. Queda por concretar si se podría utilizar *plugins* para integrarlos de manera eficiente. No es prioritario, aunque se considerará para mejorar la facilidad de uso de la plataforma.
  - Casos de uso: Integración de servicios web externos.
 

La plataforma visualizará este tipo de contenidos interactuando con los servicios externos.
  - Asociaciones: Visualización de la plataforma, Visualización de la plataforma con



dispositivos móviles.

- El acceso a la misma se deberá integrar a su vez con Twitter y Facebook en los comentarios y el acceso como usuario al sitio. Teniendo en cuenta la necesidad de mantener también el sistema de gestión de usuarios propio.
  - Definición: Se acuerda que el acceso a la plataforma deberá generar un usuario en la plataforma pero que podrá accederse con las cuentas de Twitter y Facebook. Esto requerirá de integración de *plugins* en la plataforma y interacción con estas APIs. El usuario una vez identificado o cuando quiera a hacer un comentario (tendrá la opción de identificarse en ese momento) interactuará con la plataforma tanto en la administración como en la visualización como sus permisos le especifiquen.
  - Casos de uso: Identificación en la plataforma.

El usuario se identifica con la cuenta de usuario de la plataforma, una cuenta Twitter o Facebook y se le asigna un nombre de usuario dentro de la plataforma. Podrá interactuar según los permisos de este usuario:

- Administrador global: gestión total de la plataforma.
- Administradores: gestión total sobre la web específica.
- Redactores: gestión de contenido de la web específica y ciertas funciones funcionales (que afectan a la integridad de la plataforma). Auditables por otros redactores y el administrador.
- Colaboradores: envío y gestión de los contenidos enviados, auditables por los demás roles.
- Usuarios: Sólo podrán visualizar y comentar en la plataforma. Se piensa en la posibilidad no definida todavía que opcionalmente puedan enviar noticias. Que éstas sí, deberán ser obligatoriamente auditadas.

Asociaciones: Gestión de la plataforma, Edición de contenido.

- La administración de la plataforma deberá poder realizarse a través de un navegador web y de manera intuitiva.
  - Definición: Wordpress provee de una parte de administración que tiene en cuenta estos aspectos. Concretamente tiene un menú a la izquierda donde se encuentran las acciones que pueden llevar a cabo los usuarios (condicionada a su rol). En la parte derecha se

despliegan estas acciones. Cada rol de usuarios puede realizar en según sus respectivos permisos unas acciones u otras.

- Casos de uso: Gestión de la plataforma

El usuario se identifica en la parte de administración de la plataforma y accede al la gestión de la misma. Según el tipo de usuario se le asignarán unas funcionalidades u otras. Especificadas en el caso “Identificación el la plataforma”.

- Asociaciones: Identificación en la plataforma, Edición de contenido.

- El contenido de la plataforma sólo podrá ser modificado por aquellas personas autorizadas para ello. La plataforma deberá de aportar diferentes roles que serán especificados más adelante en la sección.

- Definición: Hablando con el administrador y los redactores se acuerda la necesidad de definir diferentes roles.

- Casos de uso: Gestión de contenido.

El usuario se identifica en la parte de administración de la plataforma y accede al la gestión de la misma. En la parte izquierda tendrá los menús que le permiten acceder a la gestión del contenido y en cada uno de ellos la posibilidad de editarlos utilizando el editor.

- Asociaciones: Identificación en la plataforma, Gestión de la plataforma.

- La gestión del contenido de la plataforma deberá poder ser realizada por una persona no técnica, es decir, que no tenga conocimientos de HTML, JavaScript, etc., de forma fácil e intuitiva. Para lo que será necesario un editor WYSIWYG (What You See Is What You Get).

- Definición: La gestión del contenido está perfectamente integrada en la parte de administración de la plataforma, con uno de los mejores editores WYSIWYG que existe.

- Casos de uso: idem anterior.

- Asociaciones: idem anterior.

- Gestión de encuestas y posibilidad de añadirla a la página principal.

- Definición: El administrador y los editores podrán crear encuestas que podrán ser mostradas en las columnas de la página principal. Se trata de manera separada porque la plataforma no trae consigo esta funcionalidad. Se deberá integrar de manera adecuada

con la “Edición de contenidos”.

- Casos de uso: Creación de encuestas

Desde la parte de administración el administrador y los editores crean la encuesta. Esta luego se visualiza en cualquiera de las columnas que se elija. Para que luego los usuarios de la página la visualicen y puedan interactuar con ella.

- Asociaciones: Identificación en la plataforma, Gestión de la plataforma, Edición de contenido.

- Gestión de eventos y posibilidad de añadirlos a la pagina principal.

- Definición: El administrador y los editores podrán crear eventos que podrán ser mostradas en las columnas de la página principal. Se trata de manera separada porque la plataforma no trae consigo está funcionalidad. Se deberá integrar de manera adecuada con la “Edición de contenidos”.

- Casos de uso: Creación de eventos.

Desde la parte de administración el administrador y los editores crean el evento. Este luego se visualiza en cualquiera de las columnas que se elija. Este bloque mostrará los últimos eventos programados.

- Asociaciones: Identificación en la plataforma, Gestión de la plataforma, Edición de contenido.

- Creación de blogs dentro de la plataforma.

- Definición: La plataforma dará la opción a crear blogs. Estos blogs serán creados por el administrador y se le otorgará al usuario correspondiente los permisos de administración para ese blog en concreto. Estos blogs tendrán las mismas características generales (que no específicas) que la página principal, pero la administración caerá a cargo de los administradores de cada uno de los mismos blogs.

- Casos de uso: Creación de blogs

Se le creará el blog al usuario que podrá gestionarlo con privilegios de administrador en ese mismo blog.

- Asociaciones: Identificación en la plataforma, Gestión de la plataforma.

- La licencia de uso del software de gestión de contenidos sera GNU/GPL o compatible.

- Definición: Al ser todo el desarrollo basado en Wordpress la licencia será GNU/GPL por lo que se asume como algo estructural.
- Casos de uso: no aplica.
- Asociaciones: todo
- La licencia de uso de la base de datos debe ser GNU/GPL o compatible.
  - Definición: Al ser MySQL la licencia será GNU/GPL por lo que se asume como algo estructural.
  - Casos de uso: no aplica.
  - Asociaciones: no aplica.
- La licencia de uso del sistema operativo del servidor web debe ser GNU/GPL o compatible.
  - Definición: Las opciones barajadas entran dentro de este supuesto ya que serán GNU/Linux y el servidor web será Apache con su licencia Apache compatible con la propuesta, por lo que se usará GNU/GPL por lo que se asume como algo estructural.
  - Casos de uso: no aplica.
  - Asociaciones: no aplica.
- Los manuales, código y documentación generados deberán de ser lo menos restrictivas posibles.
  - Definición: El código que generemos será también GNU/GPL, lo mismo con la documentación y manuales que generemos. Se analizará en caso de la documentación y manuales la posibilidad de utilizar otro tipo de licencias más acordes con el contenido como puede ser alguna versión de las licencias Creative Commons. En concreto con las condiciones by-sa.
  - Casos de uso: no aplica.
  - Asociaciones: no aplica.

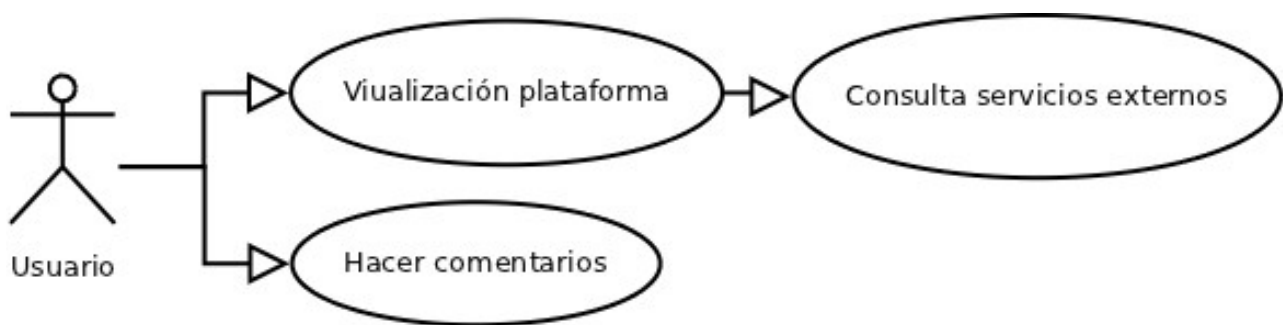
Quedan fuera de estas descripciones todos aquellos casos que implemente Wordpress y que no tengan relación directa con la implementación de la plataforma. Algunos de los cuales se han mencionado en estas líneas, como los niveles de usuario, pero por expresa necesidad a la hora de definir los casos de uso de la plataforma en concreto. Todo lo que aporte la base tecnológica de

Wordpress está bastante bien documentado en [http://codex.wordpress.org/Developer\\_Documentation](http://codex.wordpress.org/Developer_Documentation) .

## 2.2.2. Diagramas de los principales casos de uso

En este apartado se definen los casos más importantes de uso aparecidos, con sus respectivos diagramas. Algunos de ellos se comentan, aunque no aparezca su diagrama por su parecido con otro.

### Visualización de la plataforma



*Ilustración 5: Diagrama del caso de uso visualización de la plataforma*

> **Visualización de la plataforma en dispositivos móviles:** Este caso de uso es igual que el anterior, con la única diferencia del dispositivo del cual se accede.

## Integración de servicios externos en la plataforma

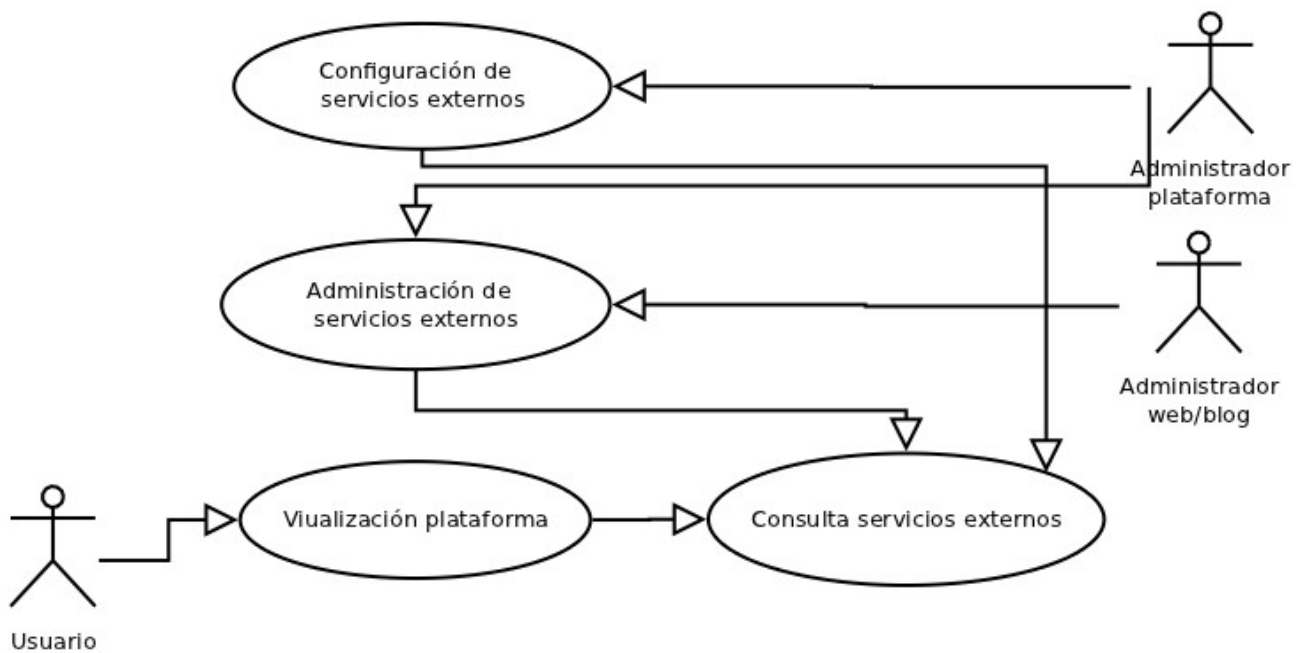


Ilustración 6: Diagrama del caso de uso integración de servicios externos en la plataforma

## Identificación en la plataforma

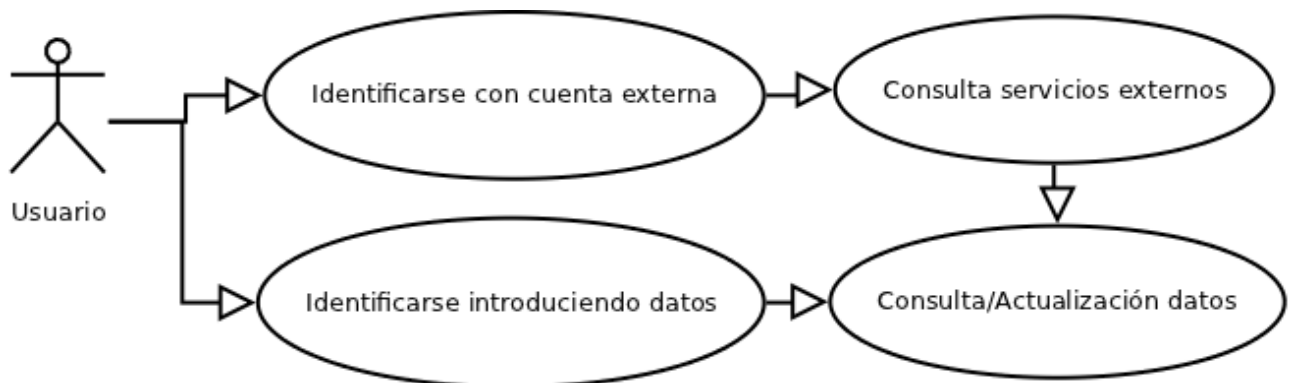


Ilustración 7: Diagrama del caso de uso identificación en la plataforma.

## Gestión de la plataforma

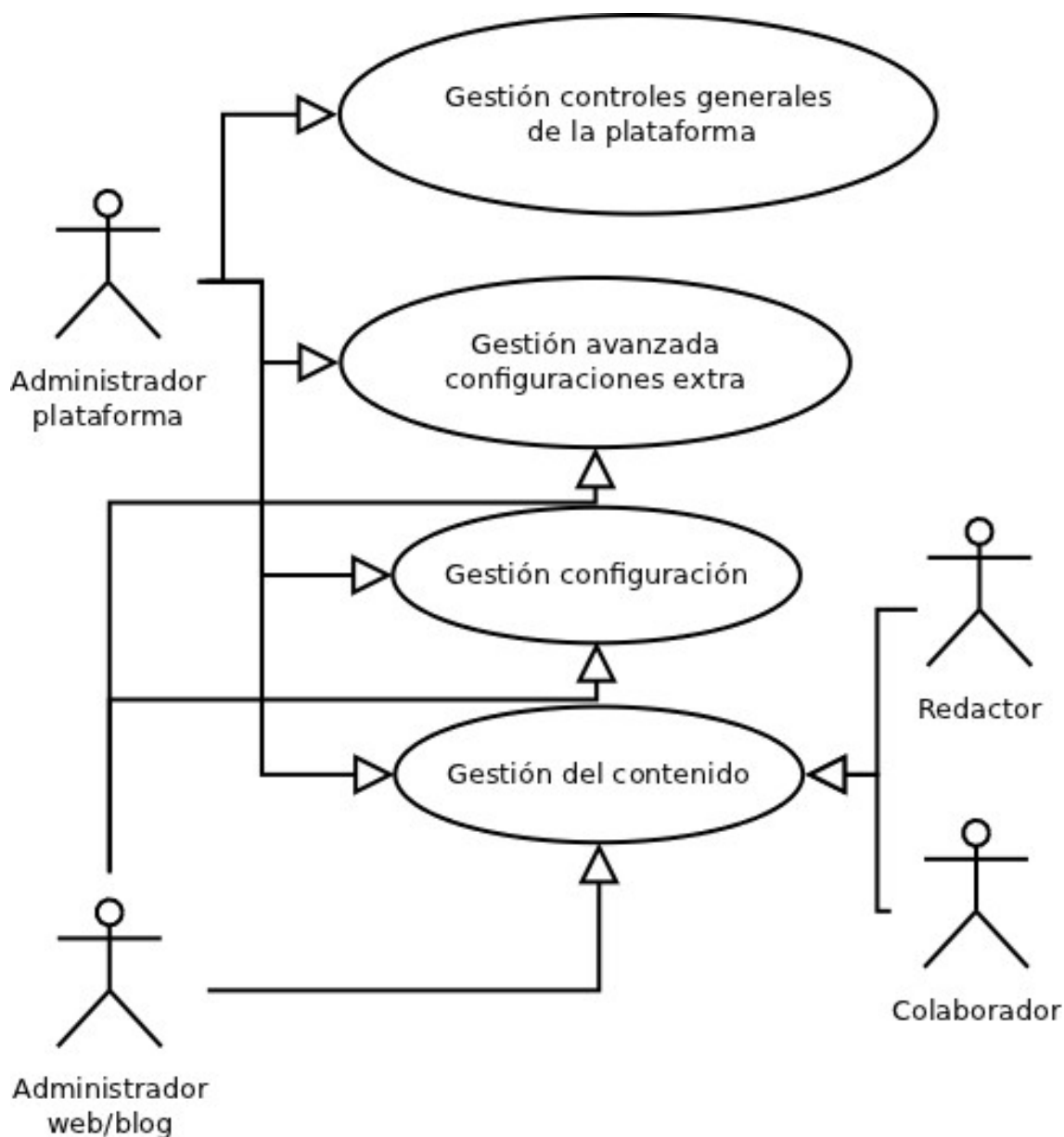


Ilustración 8: Diagrama del caso de uso gestión de la plataforma

> **Gestión de contenido:** La gestión de contenido (creación, edición, actualización y eliminación) y los actores que actúan quedan bien descritos en el diagrama anterior. Los casos de **Gestión de eventos** y **Gestión de encuestas** como subcasos de la gestión de contenidos que no requieren de más explicación.

**Creación de blogs:** Aunque parte de la gestión de la plataforma por ser especial esta parte se trata de distinta manera, por lo que es interesante crear un diagrama distinto.



*Ilustración 9: Diagrama del caso de uso creación de blogs*

## **2.3. Definición de interfaces de usuario**

En este apartado definiremos los interfaces de usuario, los diferentes perfiles y concretaremos un poco más los interfaces de usuario de cada caso de uso.

### **2.3.1. Perfiles de usuario**

La plataforma será utilizada por diferentes tipos de usuarios, que el sistema tomará en cuenta y a los que se le asignarán una serie de permisos y que normalmente tendrán ciertas características.

#### **Administrador de la plataforma**

Tiene los permisos definidos para el “Administrador global” del caso de uso “Identificación en la plataforma”. En general tendrá estas características:

- Con ciertos conocimientos técnicos.
- Acostumbrado a administrar gestores de contenido.
- Conocimientos analítico y acostumbrado a la toma de decisiones.
- Sin conocimientos periodísticos excesivos, a la hora de abordar esta temática.

#### **Administrador/es de página principal/blog**

Tiene los permisos definidos para el “Administrador” del caso de uso “Identificación en la plataforma”. En general tendrá estas características:

- Ciertos conocimientos de algunas funcionalidades en gestores de contenido.
- Conocimientos básicos de funcionamiento de la plataforma a administrar.
- Acostumbrado a utilización de editores online, correo electrónico y herramientas online (y



ofimática/edición offline).

- Con conocimiento a nivel de contenido de la temática en concreto a tratar en los contenidos.

### **Redactores de página principal/blog**

Tiene los permisos definidos para el “Redactor” del caso de uso “Identificación en la plataforma”.

En general tendrá estas características:

- Ciertos conocimientos de edición online de contenidos, pero de manera no técnica.
- Conocimientos únicamente en la edición de contenidos.
- Acostumbrado a utilización de editores online, correo electrónico y herramientas online (y ofimática/edición offline).
- Con conocimiento a nivel de contenido de la temática en concreto a tratar en los contenidos.

### **Colaboradores de página principal/blog**

Tiene los permisos definidos para el “Colaborador” del caso de uso “Identificación en la plataforma”. En general tendrá estas características:

- Ciertos conocimientos de edición online de contenidos, pero de manera no técnica.
- Conocimientos únicamente en la redacción de contenido propio.
- Acostumbrado a utilización de editores online, correo electrónico y herramientas online (y ofimática/edición offline).
- Con conocimiento a nivel de contenido de la temática en concreto a tratar en los contenidos.

### **Usuarios de la plataforma**

Tiene los permisos definidos para el “Usuarios” del caso de uso “Identificación en la plataforma”.

En general tendrá estas características:

- No se le presume conocimiento técnico alguno.
- Se puede considerar que la utilización de editores online, correo electrónico y herramientas online (y ofimática/edición offline) es común en este perfil aunque no siempre.

### 2.3.2. Principios generales de la interface de usuario

Estos son los principios generales que tendrá la plataforma. En todos los casos se intentará acercarse a esta propuesta aunque en la fase de diseño pueden tomarse otras modificaciones.

- El acceso a la aplicación y su uso se realizará a través de un navegador web.
- Será *multisite* por lo que con la misma plataforma se podrán crear más una web. La principal y los blogs. Cada uno tendrá sus propia visualización aunque compartirán la administración.
- Esta parte de administración incluirá en su parte izquierda los controles de mando para acceder a las diferentes funciones. En la parte derecha se desplegarán los formularios respectivos a esa funcionalidad. Esta parte se adaptará a los permisos que tengan cada uno de los usuarios.
- La edición y creación de los diferentes tipos de contenido (encuestas, eventos...) se hará mediante formularios sencillos. En el caso de que lo requiera, se utilizará un editor WYSIWYG.
- Todos los interfaces serán intuitivos y utilizarán AJAX para hacer más ágil y visual la administración de la plataforma.
- Los avisos y actualizaciones de software de la plataforma se mostrarán en rojo en la parte de arriba del menú de administración.

### 2.3.3 Interfaces de usuario de casos de uso

Estos son los interfaces de usuario más importantes respecto a los casos de uso definidos.

#### Visualización de la plataforma

Esta es la interface de la plataforma. De su página principal a la izquierda y de las páginas de contenidos a la derecha.

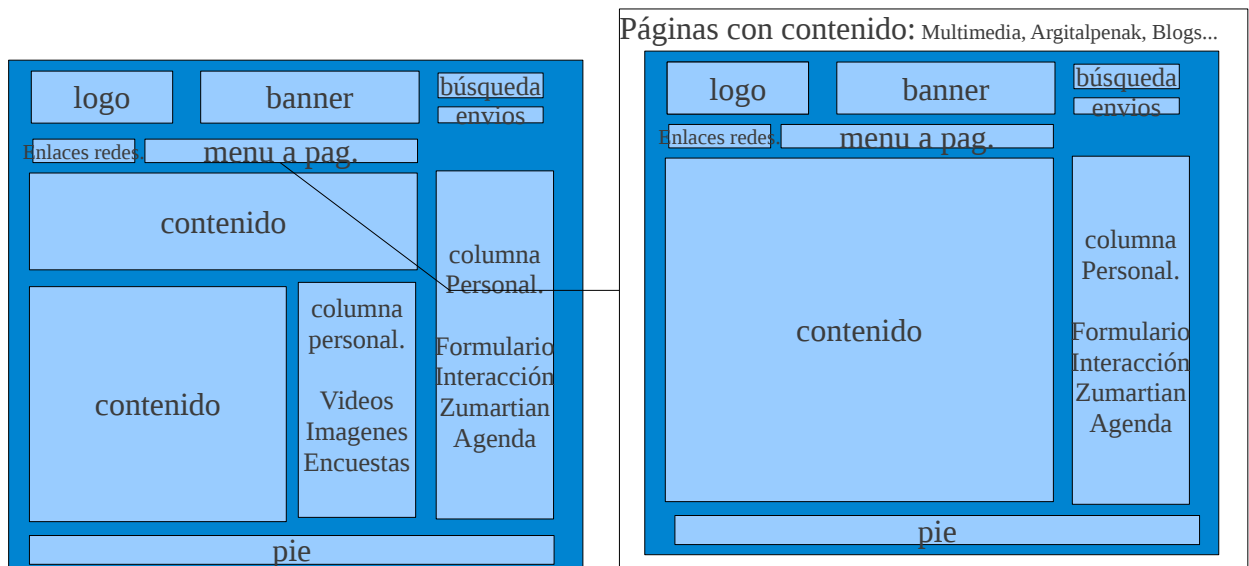


Ilustración 10: Interface de la visualización general de la plataforma

#### Visualización de la plataforma en dispositivos móviles

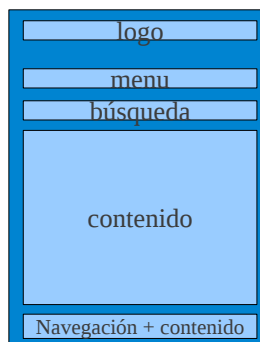


Ilustración 11: Interface de la visualización móvil de la plataforma

## Integración de contenido de servicios web externos

En rojo donde se integrará el contenido de servicios web externos. Estos elementos irán en bloques en color amarillo (*widgets*) en las columnas que se muestran pudiendo moverse.

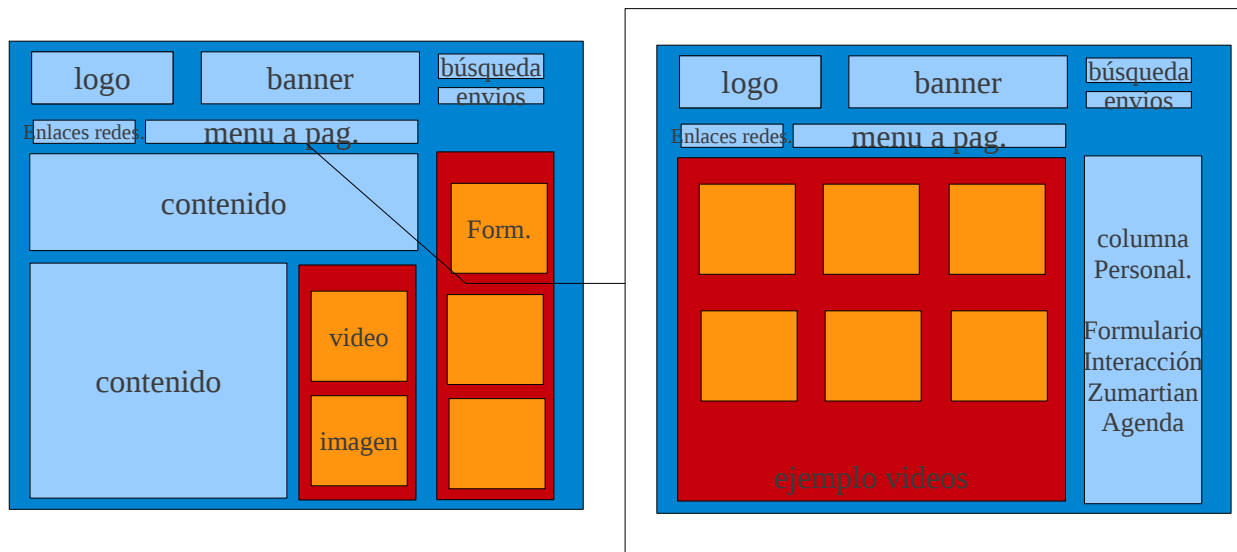


Ilustración 12: Interface de la Integración de contenido de servicios web externos

## Identificación en la plataforma

A la izquierda se muestra la entrada a la parte de administración. En la derecha la opción de los usuarios a hacer comentarios, para lo que se pueden identificar desde la parte visual de la plataforma.

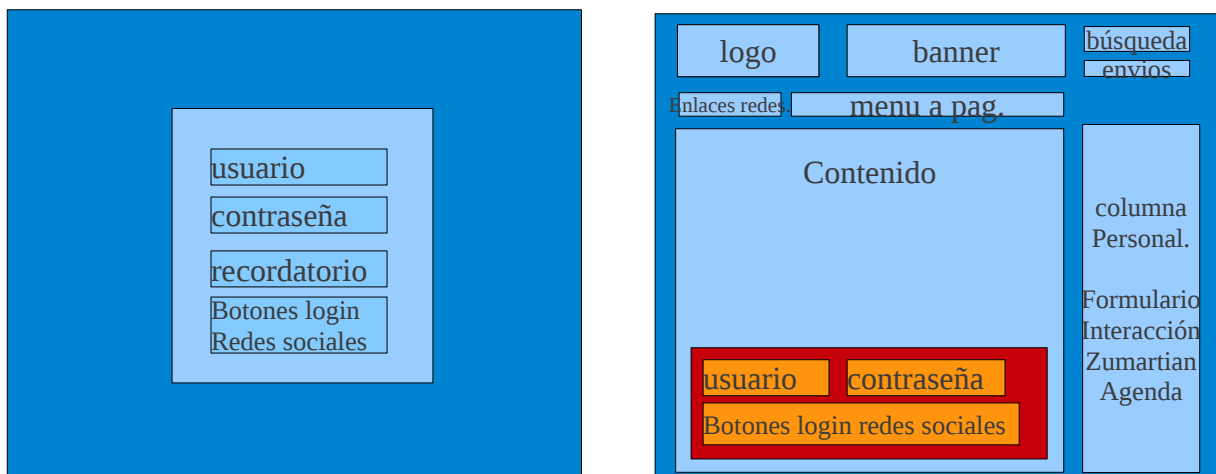
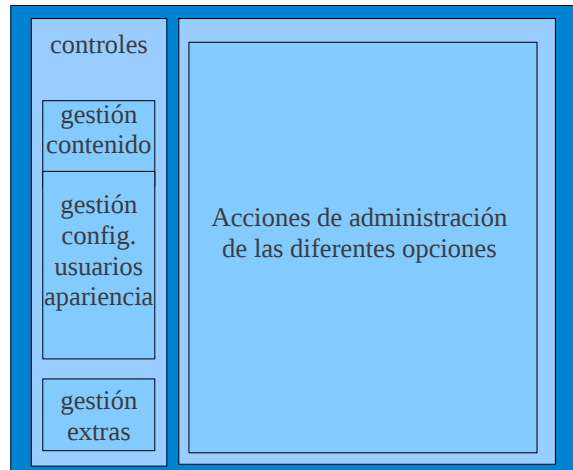


Ilustración 13: Interface de la identificación en la plataforma

## Gestión de la plataforma

Esta vista es la parte de administración de la plataforma. Se diferencian claramente la parte de

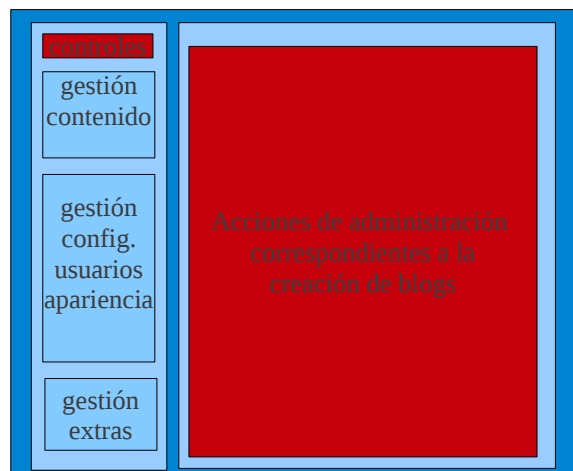
controles y la de las acciones de administración que serán diferentes según los controles.



*Ilustración 14: Interface de la gestión de la plataforma*

### **Creación de blogs**

Es una parte concreta de la gestión de la plataforma. Se activa al utilizar el control de esa acción. Es parecido para todos los controles.



*Ilustración 15: Interface de la creación de blogs*

### **Gestión de contenido**

Es una parte concreta de la gestión de la plataforma. Aunque los contenidos no son iguales, en general sería parecido para los demás. Se activa al utilizar el control de gestión de contenido correspondiente. Aquí se hace referencia al uso de un editor WYSIWYG.

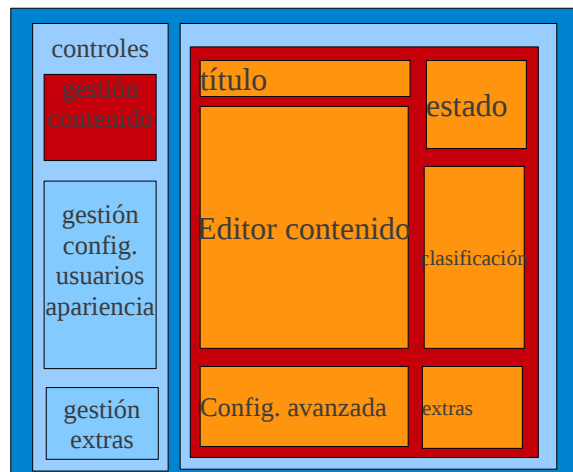


Ilustración 16: Interface de la gestión de contenido

- **Gestión de eventos**

Es una parte concreta de la gestión de contenido. El interface sería parecido con pequeños matices al de edición de contenido.

- **Gestión de encuestas**

Es una parte concreta de la gestión de contenido. El interface sería parecido con pequeños matices al de edición de contenido.

Nota: la visualización de los blogs queda fuera de este estudio. Debido a que desde el principio del proyecto se considera que quedará tal y como está en la actualidad. Su administración y la de la página principal de la plataforma a su vez son las mismas. Por eso los interfaces que aparecen aquí tienen más que ver con la página principal (que se coje como ejemplo) para visualizar todos los interfaces de la plataforma.

## 2.4. Especificación del plan de pruebas

En este apartado se definirán las pruebas que habrá que llevarse a cabo para el buen funcionamiento y la aceptación del sistema. Se realizarán diferentes pruebas unitarias, integración, sistema, implantación y aceptación. Analizando su alcance y requisitos.

### Prueba unitaria de la plataforma (sin los módulos).

- Usuarios: Administrador del sistema y Administradores de la web principal y blogs.
- Entregable: La prueba realizará una secuencia de pruebas y constatará la falta de errores y el buen funcionamiento de la plataforma en sus inicios. Se pondrán a prueba las funciones de

administración junto con la creación de algún contenido, aunque en esta fase sólo se quiere analizar la correcta instalación de la plataforma. Sobre todo porque se considera que la base tecnológica utilizada, Wordpress, es bastante solida.

- Aceptación: La prueba se considerará valida si las webs pueden ser visualizadas y la parte de administración es accesible y funcional.
- Requisitos: Disponer la aplicación instalada en un servidor web de desarrollo con los requisitos requeridos por la misma accesible para ser probados.

### **Prueba unitaria de cada uno de los módulos (plugins) / desarrollos añadidos al sistema.**

- Usuarios: Administrador del sistema y Administradores de la web principal y blogs.
- Entregable: La prueba realizará una secuencia de pruebas y constatará la falta de errores y el buen funcionamiento de cada modulo. Para lo cual se testearán las funcionalidades del módulo desde la parte de administración, viendo cuando esto sea necesario que incidencia tiene en la parte de visualización de la plataforma. Para que las pruebas sean correctas sólo la plataforma y el módulo en cuestión deben de estar en funcionamiento. Todos los demás módulos deben de estar desactivados.
- Aceptación: La prueba se considerará valida si no hay errores visibles en la plataforma tras la instalación del módulo y las funcionalidades que añade funcionan correctamente
- Requisitos:
  - Disponer la aplicación instalada en un servidor web de desarrollo con los requisitos requeridos por la misma accesible para ser probados.
  - Disponer a su vez del módulo en cuestión en el servidor y que este este activo.

### **Prueba de integración de los desarrollos/módulos en la plataforma.**

- Usuarios: Administrador del sistema y Administradores de la web principal y blogs.
- Entregable: La prueba realizará una secuencia de pruebas y constatará la falta de errores y el buen funcionamiento de los módulos. Para lo cual se testearán las funcionalidades de los módulo desde la parte de administración, viendo cuando esto sea necesario que incidencia tiene en la parte de visualización de la plataforma. Se repetirán las comprobaciones de las pruebas unitarias de los módulos, ahora en un contexto dónde todos ellos estén activos.
- Aceptación: La prueba se considerará valida si no hay errores visibles en la plataforma tras

la instalación del módulo y las funcionalidades que añada funcionan correctamente

- Requisitos:
  - Disponer la aplicación instalada en un servidor web de desarrollo con los requisitos requeridos por la misma accesible para ser probados.
  - Disponer a su vez del los módulos en cuestión en el servidor y que este estén activos.

### **Prueba de sistema de funcionamiento de toda la plataforma.**

- Usuarios: Administrador del sistema, Administradores de la web principal y blogs. Un subgrupo de Redactores, Colaboradores y Usuarios.
- Entregable: Se analizarán las funcionalidades que el subgrupo de redactores, colaboradores y usuarios puedan realizar. Las pruebas serán las mismas debido al nivel de permisos algunos sólo podrán realizar algunas, siendo los redactores en este caso los que más pruebas realizaran. Estarán basadas sobre todo en la creación y edición de contenido. Además se analizará también la interacción y la visualización de los cambios en la visualización de la página.
- Aceptación: La prueba se considerará válida si no hay errores visibles en la plataforma. Si todas las acciones han podido llevarse a cabo y la aceptación ha sido generalizada. Si no es así se realizarán modificaciones y posteriores pruebas.
- Requisitos:
  - Disponer la plataforma instalada en un servidor web de desarrollo con los requisitos requeridos por la misma accesible para ser probados.
  - Disponer todos los módulos en marcha y en funcionamiento para poder probar todas y cada una de las funcionalidades.

### **Prueba de implantación de la plataforma en entorno real.**

- Se repetirá la prueba anterior en el servidor de producción. Para lo cual será necesario tener preparado también este entorno, con un servidor web y todo los requisitos necesarios.

### **Prueba de aceptación de la plataforma por parte de los usuarios.**

- Usuarios: Administrador del sistema, Administradores de la web principal y blogs. Todos los redactores, colaboradores y usuarios.
- Entregable: Se probará la plataforma en sus funciones habituales a todo el grupo de usuarios



implicado durante un periodo de tiempo. En este periodo de tiempo cada usuario testará la plataforma desde su realidad habitual. Se recogerán las percepciones, errores y aportaciones que se realcen en este periodo de tiempo, para solucionar o mejorar la plataforma a futuro.

- Aceptación: La prueba se considerara valida si el nivel de aceptación general de la plataforma es optimo. Si las funcionalidades no producen errores que impidan su correcto funcionamiento y si las mejoras propuestas no son imprescindibles a corto plazo.
- Requisitos:
  - Disponer la plataforma instalada en un servidor web de producción con los requisitos requeridos por la misma accesible para ser probados.
  - Comunicación con los usuarios de la plataforma para poder recoger su *feedback*.

### 3. Diseño

En esta fase obtendremos los modelos y especificaciones que lo definen a partir del análisis realizado en la fase anterior. Las actividades que llevemos a cabo en esta fase nos permitirán determinar las especificaciones de desarrollo e integración, así como definir el entorno de pruebas e implantación necesarios para su correcto funcionamiento.

#### 3.1. Arquitectura

En esta fase identificaremos los componentes que formarán el sistema.

##### 3.1.1. Definición de niveles de arquitectura

Para expresar la arquitectura de nuestra plataforma, usamos la notación UML en los diagramas y tarjetas CRC (Clase-Responsabilidad-Colaborador).

En este primer diagrama se muestran los diferentes componentes y sus relaciones, utilizando UML.

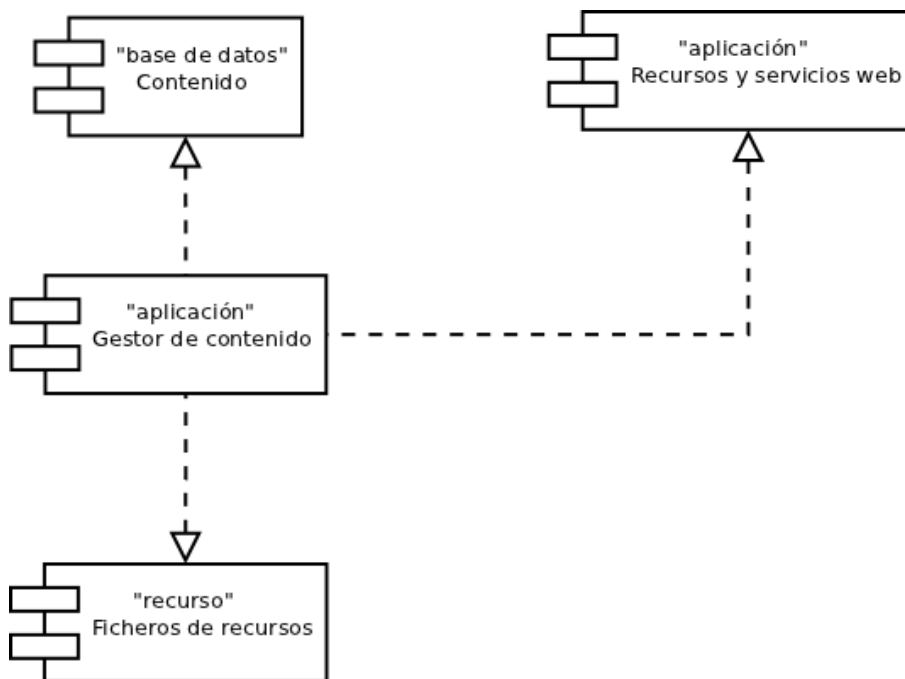


Ilustración 17: Diagrama UML general de la arquitectura

En este segundo diagrama se profundiza añadiendo los interfaces de comunicación de los mismos para tener más detalle. Se detallarán a su vez mediante tarjetas CRC las capacidades de cada uno de ellos más abajo.

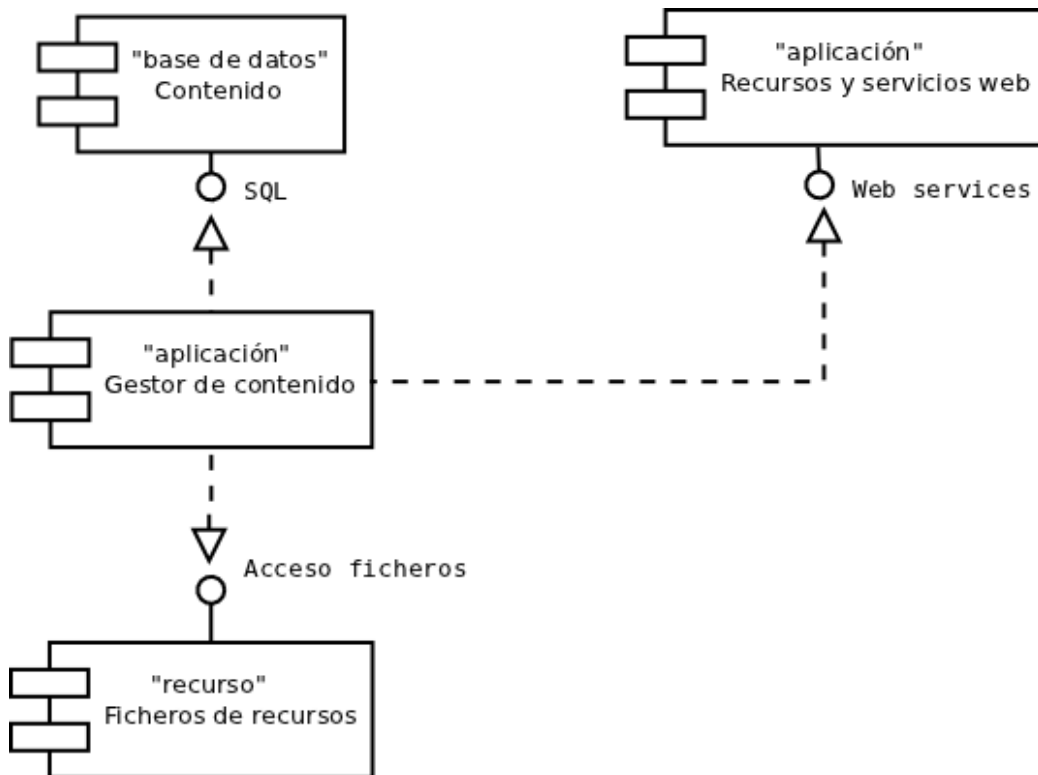


Ilustración 18: Diagrama UML general de la arquitectura con sus interfaces

Estos son las tarjetas CRC de los componentes más importantes.

Gestor de contenido	
Permite administrar el contenido del sitio web.	<ul style="list-style-type: none"> <li>• Base de datos de contenido.</li> <li>• Ficheros de recurso.</li> <li>• Recursos y servicios web.</li> </ul>
Muestra el contenido público del sitio web.	
Muestra el contenido externo del sitio web.	

Recursos y servicios web	
Permite identificarse en la plataforma con servicios web externos.	<ul style="list-style-type: none"> <li>• Gestor de contenido.</li> </ul>
Permite integrar recursos de servicios web externos (videos, imagenes, documentos) en la plataforma.	

### 3.1.2. Especificación de estándares, normas de diseño y construcción

Los documentos generados de aquí adelante deberán de cumplir esta serie de normas. Debido a la

escasa complejidad del grupo técnico no se requiere de grandes infraestructuras, aún así estas son las pautas más importantes.

**Documentos de diseño:** Se acuerda utilizar un wiki, como herramienta de documentación. Este wiki deberá de ser capaz de exportar a otros formatos, necesariamente a OpenDocument y PDF para su consulta. En el wiki quedará registrado :

- Título del documento.
- El creador del mismo
- Lista de autores que han intervenido y la fecha de su primera intervención.
- Lista resumida de cambios introducidos en el documento a medida que se vayan produciendo (cambio, fecha y autor).

• **Diagramas de diseño:** para los diagramas de diseño se acuerda usar la notación Unified Modeling Language, Lenguaje Unificado de Modelado es el lenguaje de [modelado](#) de sistemas de [software](#) más conocido y utilizado en la actualidad; está respaldado por el [OMG](#) (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

• **Documentación técnica:** la documentación técnica será posiblemente la que más revisiones sufrirá y contendrá también enlaces a documentaciones de las herramientas usadas, especificaciones de programación (API), etc. Se propone utilizar también otro wiki para estos cambio. Combinándolo con referencias al código, o incluso trozos de código que puedan ser importados directamente.

### 3.1.3. Identificación de subsistemas

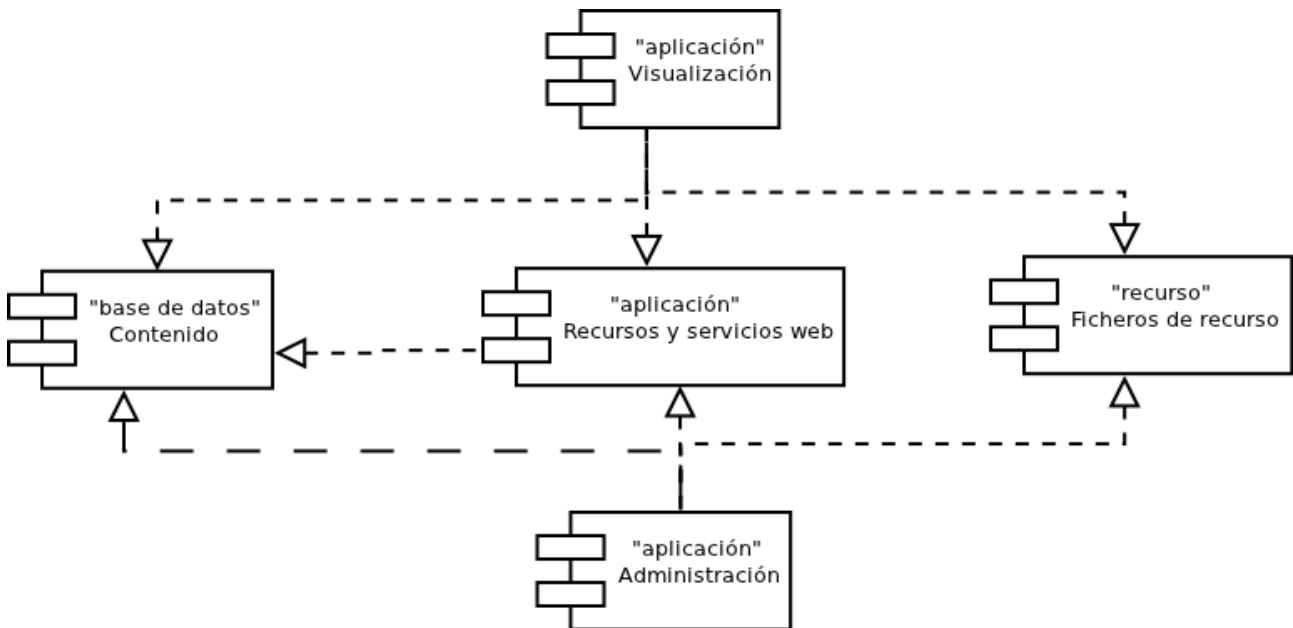
En este apartado se diferencian los diferentes subsistemas de la plataforma, para reducir la complejidad que implicaría detallar todo el sistema.

En este caso se detallan tres subsistemas:

- Subsistema de visualización: Es la parte visible de la plataforma que se concreta en las diferentes webs existentes (página web principal y blogs).
- Subsistema de administración: Es la parte en la que se gestionan los controles, el contenido y toda la configuración de la plataforma.
- Subsistema de recursos y servicios web externos: Se puede hacer aquí dos subdivisiones:

- Servicios identificación: Estos servicios web sirven para identificarse en la plataforma.
- Servicios de recursos: Estos servicios web sirven para utilizar recursos externos dentro de la plataforma.

En este diagrama se muestran los tres subsistemas. Los **subsistemas de visualización y administración** son bastante parecidos en sus interacciones que quedan bastante bien descritas en este diagrama.



*Ilustración 19: Diagrama UML de los subsistemas de visualización y administración*

Por otra parte el subsistema **Recursos y servicios web externos** requiere de más explicación y está es la que se describe en el siguiente diagrama. Se extiende primero el anterior diagrama y luego se muestra a su vez los dos subsistemas de identificación y recursos de servicios web externos de manera separada al sistema.

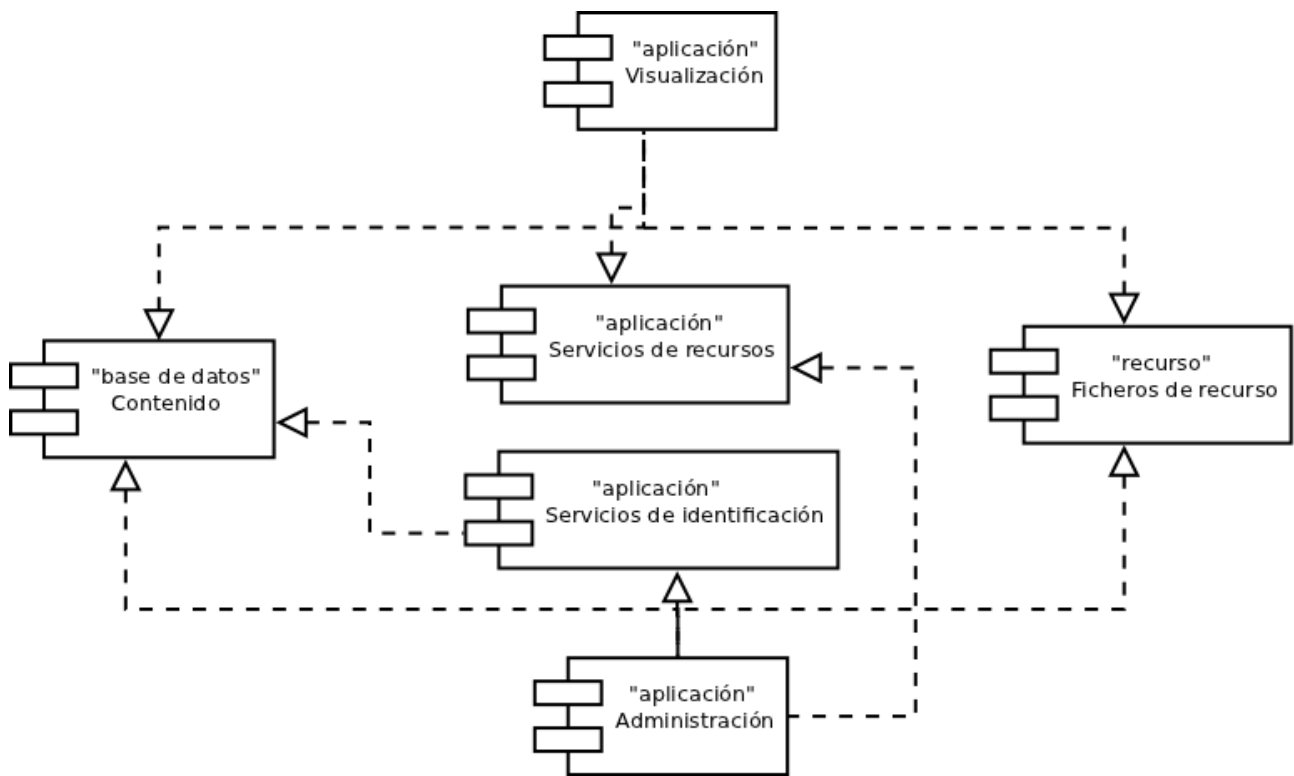


Ilustración 20: Diagrama UML del subsistema de recursos y servicios web externos desplegado

Este es el diagrama de Servicios de recursos.

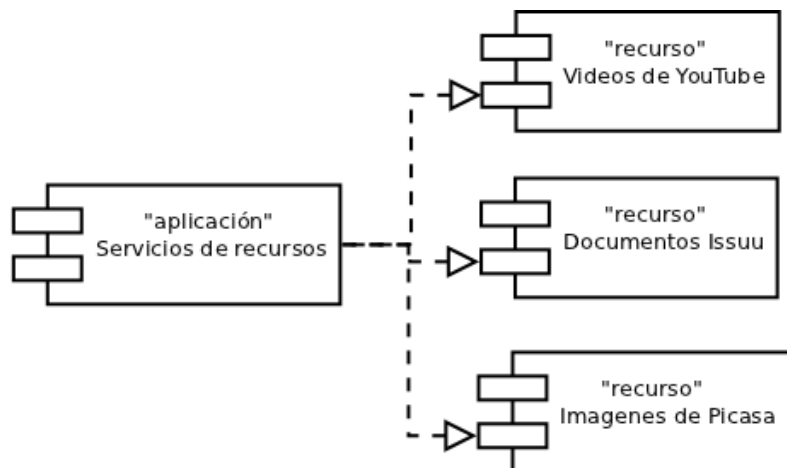
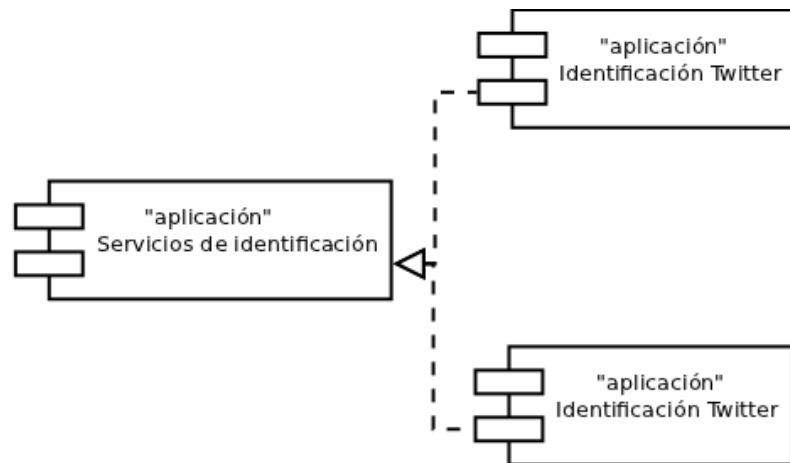


Ilustración 21: Diagrama UML del subsistema de servicios de recursos

Este es el diagrama de Servicios de identificación.



*Ilustración 22: Diagrama UML del subsistema de servicios de identificación*

### 3.2 Revisión de casos de uso

En esta fase estableceremos las características, revisaremos los requisitos y diseñaremos las clases de todo el sistema respecto a los casos de uso.

#### 3.2.1 Revisión de los subsistemas según los casos de uso

Se revisan los casos de uso relevantes, analizando en los subsistemas que intervienen en el mismo, cuáles serán los objetos que componen cada subsistema y sus interacciones. Algunos casos como la “integración de servicios externos” se analizan dentro de otros casos por estar enteramente integrados en más de un caso.

##### **Caso de uso: Visualización de la plataforma (idem Visualización de la plataforma móvil)**

El caso de uso referido a la visualización de la plataforma está relacionado directamente con el **subsistema de visualización**. La correlación entre este caso y este subsistema es obvia, permite la visualización de la plataforma mediante la web. Pero además tiene relación con los siguientes subsistemas:

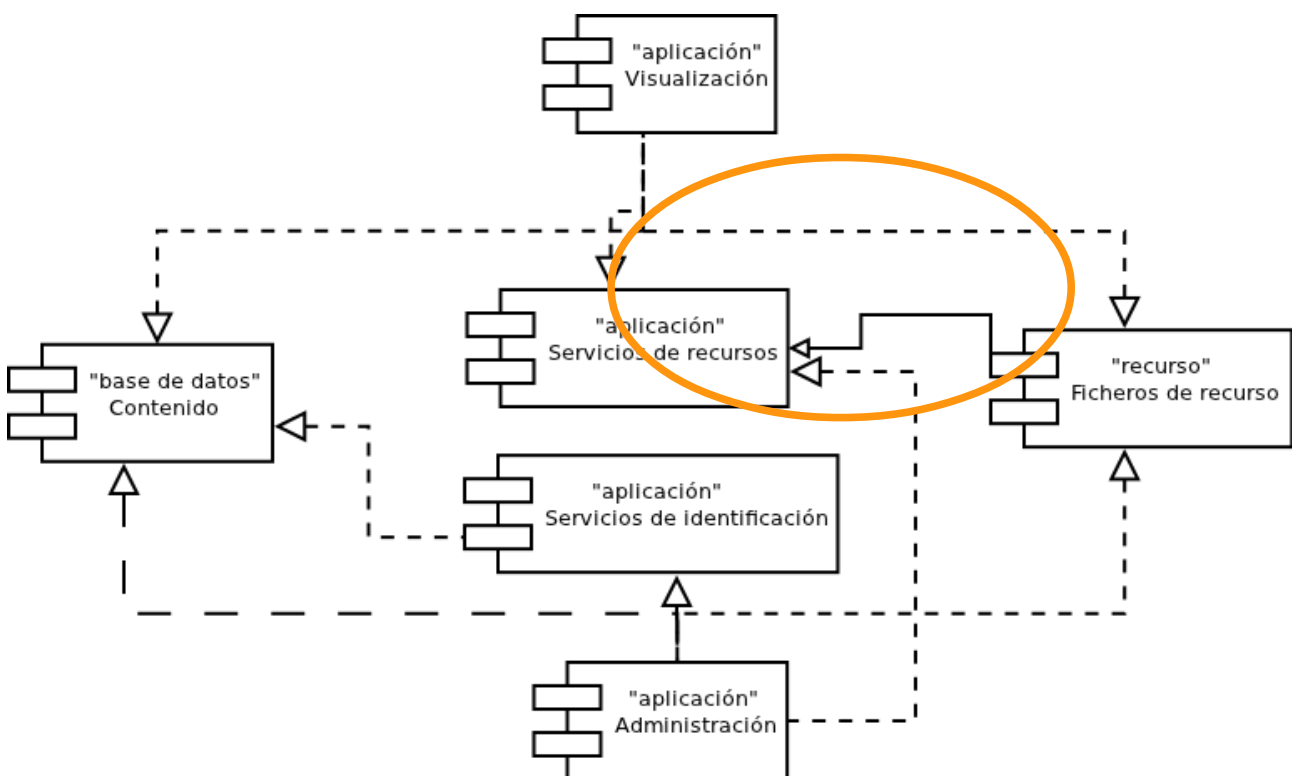
- Subsistema servicios de recurso: Consultará los servicios web externos para visualizar los recursos (videos de YouTube, imágenes de Picasa, documentos de Issuu...).
- Recurso: Recuperara y visualizara los ficheros de recurso que sean necesarios en cada caso.

- Base de datos: Recuperara y visualizara los datos extraídos de la base de datos que sean necesarios en cada caso.

El estudio detallado del caso de uso, y de los subsistemas que intervienen en él, da como resultado el cambio del **sistema de servicios de recurso**, para que soporten características como:

- Sincronización bidireccional de los documentos de Issuu: Ahora cuando un documento se suba mediante el sistema de gestión de recursos integrado este se coordinará con el subsistema de servicios de recurso para subirlo también al servicio web.

Esto modifica en cierta manera la relación entre clases del subsistema de servicios web de recursos, aunque en la práctica supondrá simplemente la integración seguramente de un plugin extra que cumpla con esta función. En la fase de desarrollo actual ya tenemos detectados algunos que puedan ayudarnos a integrar esta funcionalidad. Los comentaremos más adelante, aunque quizás necesiten un desarrollo a medida. Aún así esta sería la modificación a realizar en el subsistema:



*Ilustración 23: Diagrama UML del subsistema de recursos y servicios web externos desplegado y ampliado*

### **Caso de uso: Gestión de la plataforma (idem Gestión del contenido)**

El caso de uso referido a la gestión de la plataforma está relacionado directamente con el



**subsistema de administración.** La correlación entre este caso y este subsistema es obvia, permite la administración de la plataforma mediante la web (la gestión de usuarios, los recursos, la configuración y todo lo relacionado con el funcionamiento interno de la misma). Pero además tiene relación con los siguientes subsistemas:

- Subsistema servicios de recurso: Consultará los servicios web externos para gestionar los recursos (videos de YouTube, imagenes de Picasa, documentos de Issuu...). Ya sea por estar integrados en el contenido o por tener que configurar aspectos relacionados con estos por ejemplo (las claves de las APIs de los servicios integrados).
- Subsistema de Servicios de identificación: Mediante este subsistema se identificarán los usuarios en la plataforma para asignarles los debidos permisos de administración en la misma y poder acceder así a la administración web de la plataforma.
- Recurso: Recuperara y visualizara los ficheros de recurso que sean necesarios en cada caso con posibilidad de modificar los mismos.
- Base de datos: Recuperara y visualizara los datos extraídos de la base de datos que sean necesarios en cada caso con posibilidad de modificar los mismos.

El estudio detallado del caso de uso de gestión de la plataforma no necesita cambios específicos en este apartado. Esto es así porque en esta fase del desarrollo esta parte esta gestionada enteramente por la herramienta seleccionada y por lo tanto se ciñe a la misma.

### **Caso de uso: Identificación en la plataforma**

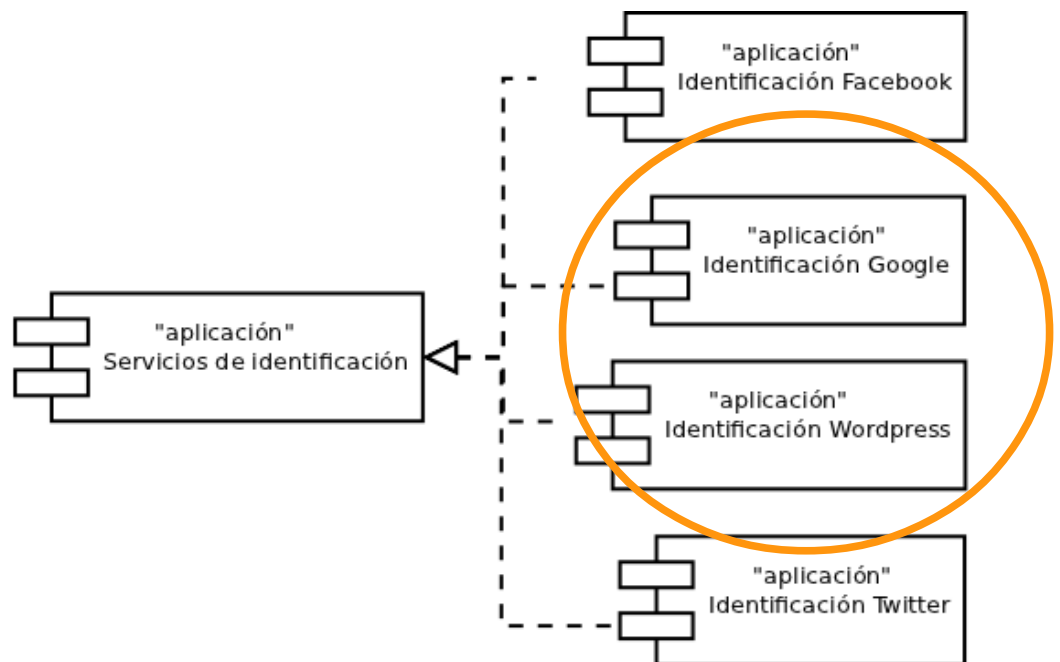
El caso de uso referido a la identificación en la plataforma está relacionado directamente con el **subsistema de servicios de identificación.** La correlación entre este caso y este subsistema es obvia, permite la identificación dentro de la plataforma. Pero además tiene relación con los siguientes subsistemas:

- Subsistema administración: Permite administrar la plataforma mediante la web, para lo cual es necesario recurrir a la identificación que proporciona el subsistema de servicios de identificación.
- Base de datos: Recuperara y visualizara los datos extraídos de la base de datos que sean necesarios en cada caso.

El estudio detallado del caso de uso de identificación en la plataforma, da como resultado el cambio

del **subsistema servicios de identificación**, de estos últimos, para que soporten características como:

- Admitir más credenciales: El análisis de casos de uso identifico dos sistemas de credenciales Twitter y Facebook. A estos habrá que añadirles otros dos como son Google y Wordpress.com. En la práctica supondrá simplemente la integración de un *plugin* que cumpla con esta función. En la fase de desarrollo actual ya tenemos detectados algunos que cumplen con esta funcionalidad, los comentaremos más adelante. Aún así este sería la modificación a realizar en el subsistema:



*Ilustración 24: Diagrama UML del subsistema de servicios de identificación ampliado*

Los casos que no se detallan están analizados dentro de los subsistemas más grandes, exceptuando el caso Identificación en la plataforma que hemos decidido tratar de manera separada.

### **3.2.2 Elección de alternativas de componentes y licencias más adecuadas**

En el estadio actual de desarrollo la mayoría de los componentes parecen ser posibles de incorporar sin tener que hacer un desarrollo a medida, exceptuando quizás la integración con Issuu. Algunos requerirán modificaciones que se irán detallando.

Para ello hemos realizado un análisis exhaustivo de los componentes más interesantes que cumplen

con los requerimientos iniciales y especificaciones de los casos de usos analizados.

El desarrollo tomará de base Wordpress (GPL) esto condiciona gran parte del desarrollo tanto a nivel técnico (uso de *plugins*, PHP sobre todo...) como a nivel de licencias (uso de licencias GPL).

Esta es la relación de software utilizado y sus correspondientes licencias:

### Entorno

Componente	Paquete	Versión	Licencia
Gestor de contenidos	Wordpress	3.3.2	GPL
Base de datos	MySQL	5.2.17	GPL
Interprete de scripts	PHP	5.1.61	GPL
Servidor web	Apache	2.2.1	Apache license
Sistema operativo	GNU/Linux	2.6.32	PHP license

### Themes y Plugins

Wordpress considera todo desarrollo en torno a la plataforma como lo son *themes* y *plugins* como obras derivadas del mismo y por lo tanto necesariamente compatibles con la GPL. Es por eso que todos los *plugins* son compatibles con esta licencia (aunque a veces necesiten utilizar servicios externos que no lo son ). Este es un extracto de la web de Wordpress donde se trata el tema de las licencias:

*The license under which the WordPress software is released is the GPLv2 (or later) from the [Free Software Foundation](#). A copy of the license is included with every copy of WordPress, but you can also read [the text of the license](#) here.*

*Part of this license outlines requirements for derivative works, such as plugins or themes. Derivatives of WordPress code inherit the GPL license. [Drupal](#), which has the same GPL license as WordPress, has an excellent page on [licensing as it applies to themes and modules](#) (their word for plugins).*

Aún así analizados los *plugin* utilizados todos son en sus respectivas versiones compatibles con la licencia GPL que pretendemos utilizar.

Wordpress en cierta manera nos obliga a utilizar la licencia GPL, que es sin duda la que queríamos utilizar ya que el cliente tampoco a puesto ninguna restricción en este caso. En el código que generemos respetaremos la licencia GPL y la integraremos dentro del paquete.

Estos son concretamente los **plugins** que pretendemos implantar en el proceso:

<b>Plugins</b>	<b>Descripción</b>
<b>Bad Behavior</b>	Deny automated spambots access to your PHP-based Web site. 2.2.6 bertsioa   Garatzailea: <a href="#">Michael Hampton</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>Contact Form 7</b>	Just another contact form plugin. Simple but flexible. 3.1.2 bertsioa   Garatzailea: <a href="#">Takayuki Miyoshi</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>Feed Template Customize</b>	This plugin modifies RSS feeds and ATOM feeds as you want. 1.0.0.1 bertsioa   Garatzailea: <a href="#">wokamoto</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>kPicasa Gallery</b>	Display your Picasa Web Galleries in a post or in a page. 0.2.9 bertsioa   Garatzailea: Guillaume Hébert   <a href="#">Bisitatu pluginaren gunea</a>
<b>Picasa Express x2</b>	Browse, search and select photos from any publicly available Picasa Web Album and add them to your post/pages. 1.5.3 bertsioa   Garatzailea: <a href="#">Wott</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>Post From Site</b>	Add a new post/page/{your custom post type} directly from your website. 3.0.1 bertsioa   Garatzailea: <a href="#">Kelly Dwan</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>Really Simple CAPTCHA</b>	Really Simple CAPTCHA is a CAPTCHA module intended to be called from other plugins. It is originally created for my Contact Form 7 plugin. 1.5 bertsioa   Garatzailea: <a href="#">Takayuki Miyoshi</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>Social Connect</b>	Allow your visitors to comment, login and register with their Twitter, Facebook, Google, Yahoo or WordPress.com account. 0.9 bertsioa   Garatzailea: <a href="#">Brent Shepherd</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>Tabber Tabs Widget</b>	Easily create a tabbed content area in your sidebar 0.38 bertsioa   Garatzailea: <a href="#">SlipFire LLC.</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>Tabber Widget</b>	Provides a simple-to-use editor to create tabbed widgets 1.0.6 bertsioa   Garatzailea: <a href="#">iThemes</a>   <a href="#">Bisitatu pluginaren gunea</a>
<b>TubePress</b>	Displays gorgeous YouTube and Vimeo galleries in your posts, pages, and/or

Plugins	Descripción
	<p>sidebar. Upgrade to <a href="#">TubePress Pro</a> for more features!</p> <p>2.4.2 bertsioa   Garatzailea: <a href="#">Eric D. Hough</a>   <a href="#">Bisitatu pluginaren gunea</a></p>
<b>Twitter Tools</b>	<p>A complete integration between your WordPress blog and <a href="#">Twitter</a>. Bring your tweets into your blog and pass your blog posts to Twitter. Show your tweets in your sidebar, and post tweets from your WordPress admin.</p>
<b>Twitter Tools - Hashtags</b>	<p>2.4 bertsioa   Garatzailea: <a href="#">Crowd Favorite</a>   <a href="#">Bisitatu pluginaren gunea</a></p> <p>Set #hashtags for blog post tweets sent by Twitter Tools. This plugin relies on Twitter Tools, configure it on the Twitter Tools settings page.</p>
<b>WP-Polls</b>	<p>2.4 bertsioa   Garatzailea: <a href="#">Crowd Favorite</a>   <a href="#">Bisitatu pluginaren gunea</a></p> <p>Adds an AJAX poll system to your WordPress blog. You can easily include a poll into your WordPress's blog post/page. WP-Polls is extremely customizable via templates and css styles and there are tons of options for you to choose to ensure that WP-Polls runs the way you wanted. It now supports multiple selection of answers.</p>
<b>WP Bannerize</b>	<p>2.63 bertsioa   Garatzailea: <a href="#">Lester 'GaMerZ' Chan</a>   <a href="#">Bisitatu pluginaren gunea</a></p> <p>WP Bannerize: an easy to use adv server with html, free text and Flash banner support.</p>
<b>WP Issuu</b>	<p>3.0.62 bertsioa   Garatzailea: <a href="#">wpXtreme</a>   <a href="#">Bisitatu pluginaren gunea</a>   <b>For more info visit <a href="#">wpXtreme Blog</a></b></p> <p>Embed Issuu publications inside a post</p>
<b>Wp NicEdit</b>	<p>2.15 bertsioa   Garatzailea: Issuu</p> <p>This plugin integrates the a very slick <a href="#">Nice editor</a> by Brian Kirchoff to wordpress comments converting it to a rich text editor.</p>
<b>WP Super Cache</b>	<p>1.0 bertsioa   Garatzailea: <a href="#">Brajesh K.Singh</a>   <a href="#">Bisitatu pluginaren gunea</a></p> <p>Very fast caching plugin for WordPress.</p>
<b>WPtouch Pro</b>	<p>1.1 bertsioa   Garatzailea: <a href="#">Donncha O Caoimh</a>   <a href="#">Bisitatu pluginaren gunea</a></p> <p>WPtouch Pro is a plugin to re-format your website with a mobile theme tailored for Apple <a href="#">iPhone</a> / <a href="#">iPod touch</a>, <a href="#">Google Android</a>, <a href="#">Blackberry Storm &amp; Torch</a> and other touch mobile devices.</p>

## Plugins

## Descripción

2.2.4 bertsioa | Garatzailea: [Dale Mugford & Duane Storey \(BraveNewCode\)](#)  
| [Bisitatu pluginaren gunea](#)

Y este es el **theme** en el que nos basaremos para trabajar sobre él. El *theme* utilizado es Magazine Basic (<http://wordpress.org/extend/themes/magazine-basic>) . Sobre el se generarán las diversas modificaciones necesarias, gracias al sistema de *child themes* ([http://codex.wordpress.org/Child Themes](http://codex.wordpress.org/Child_Themes)) de Wordpress.

### 3.2.3 Especificaciones de desarrollo y pruebas

La plataforma que componen los dos subsistemas principales de administración y visualización deberán de permitir a los usuarios gestionar los contenidos y visualizarlos de manera adecuada. Para ello utilizaremos Wordpress como base para el mismo. Para los demás subsistemas y funcionalidades añadidas se utilizarán *plugins* de Wordpress como los ya expuestos con anterioridad. En algunos de los casos se integrarán directamente y en otros requerirán modificaciones y/o desarrollo. En cualquier caso, después de terminar el desarrollo en los dos casos se les aplicaran las pruebas correspondientes.

#### Especificaciones

En los casos que se necesiten desarrollar o modificar los plugins indicados se utilizarán estas herramientas:

- PHP (<http://php.net>) como lenguaje de scripting con el añadido en algunos casos de Javascript.
- Bluefish (<http://bluefish.openoffice.nl/index.html>) o Gedit (<http://projects.gnome.org/gedit/>) cualquiera de las dos herramientas se antojan adecuadas para modificar o desarrollar.
- Documentación se utilizará una sintaxis propia que se utilizan para identificar los ficheros modificados. Se marcan con ello sólo las parte modificadas para saber exactamente cuales son los ficheros modificados. Esto más el uso de git (<http://git-scm.com/>) y la documentación en formato texto en los desarrollos que necesiten más especificaciones.

#### Pruebas

Las pruebas seguirán un plan de pruebas conciso que se recoge en las siguientes líneas.

### **Prueba unitaria de la plataforma (sin los módulos).**

- Usuarios: Administrador del sistema y Administradores de la web principal y blogs.
- Entregable: La prueba realizará una secuencia de pruebas y constatará la falta de errores y el buen funcionamiento de la plataforma en sus inicios. Se pondrán a prueba las funciones de administración junto con la creación de algún contenido, aunque en esta fase sólo se quiere analizar la correcta instalación de la plataforma. Sobre todo porque se considera que la base tecnológica utilizada, Wordpress, es bastante solida.
- Aceptación: La prueba se considerará valida si las webs pueden ser visualizadas y la parte de administración es accesible y funcional.
- Requisitos: Disponer la aplicación instalada en un servidor web de desarrollo con los requisitos requeridos por la misma accesible para ser probados.

### **Prueba unitaria de cada uno de los módulos (plugins) / desarrollos añadidos al sistema.**

- Usuarios: Administrador del sistema y Administradores de la web principal y blogs.
- Entregable: La prueba realizará una secuencia de pruebas y constatará la falta de errores y el buen funcionamiento de cada modulo. Para lo cual se testearán las funcionalidades del módulo desde la parte de administración, viendo cuando esto sea necesario que incidencia tiene en la parte de visualización de la plataforma. Para que las pruebas sean correctas sólo la plataforma y el módulo en cuestión deben de estar en funcionamiento. Todos los demás módulos deben de estar desactivados.
- Aceptación: La prueba se considerará valida si no hay errores visibles en la plataforma tras la instalación del módulo y las funcionalidades que añade funcionan correctamente
- Requisitos:
  - Disponer la aplicación instalada en un servidor web de desarrollo con los requisitos requeridos por la misma accesible para ser probados.
  - Disponer a su vez del módulo en cuestión en el servidor y que este este activo.

### **Prueba de integración de los desarrollos/módulos en la plataforma.**

- Usuarios: Administrador del sistema y Administradores de la web principal y blogs.
- Entregable: La prueba realizará una secuencia de pruebas y constatará la falta de errores y el buen funcionamiento de los módulos. Para lo cual se testearán las funcionalidades de los

módulo desde la parte de administración, viendo cuando esto sea necesario que incidencia tiene en la parte de visualización de la plataforma. Se repetirán las comprobaciones de las pruebas unitarias de los módulos, ahora en un contexto dónde todos ellos estén activos.

- Aceptación: La prueba se considerará válida si no hay errores visibles en la plataforma tras la instalación del módulo y las funcionalidades que añade funcionan correctamente
- Requisitos:
  - Disponer la aplicación instalada en un servidor web de desarrollo con los requisitos requeridos por la misma accesible para ser probados.
  - Disponer a su vez del los módulos en cuestión en el servidor y que este estén activos.

### **Prueba de sistema de funcionamiento de toda la plataforma.**

- Usuarios: Administrador del sistema, Administradores de la web principal y blogs. Un subgrupo de Redactores, Colaboradores y Usuarios.
- Entregable: Se analizarán las funcionalidades que el subgrupo de redactores, colaboradores y usuarios puedan realizar. Las pruebas serán las mismas debido al nivel de permisos algunos sólo podrán realizar algunas, siendo los redactores en este caso los que más pruebas realizaran. Estarán basadas sobre todo en la creación y edición de contenido. Además se analizará también la interacción y la visualización de los cambios en la visualización de la página.
- Aceptación: La prueba se considerará válida si no hay errores visibles en la plataforma. Si todas las acciones han podido llevarse a cabo y la aceptación ha sido generalizada. Si no es así se realizarán modificaciones y posteriores pruebas.
- Requisitos:
  - Disponer la plataforma instalada en un servidor web de desarrollo con los requisitos requeridos por la misma accesible para ser probados.
  - Disponer todos los módulos en marcha y en funcionamiento para poder probar todas y cada una de las funcionalidades.

### **Prueba de implantación de la plataforma en entorno real.**

- Se repetirá la prueba anterior en el servidor de producción. Para lo cual será necesario tener preparado también este entorno, con un servidor web y todo los requisitos necesarios.



### Prueba de aceptación de la plataforma por parte de los usuarios.

- Usuarios: Administrador del sistema, Administradores de la web principal y blogs. Todos los redactores, colaboradores y usuarios.
- Entregable: Se probará la plataforma en sus funciones habituales a todo el grupo de usuarios implicado durante un periodo de tiempo. En este periodo de tiempo cada usuario testeará la plataforma desde su realidad habitual. Se recogerán las percepciones, errores y aportaciones que se realcen en este periodo de tiempo, para solucionar o mejorar la plataforma a futuro.
- Aceptación: La prueba se considerara valida si el nivel de aceptación general de la plataforma es optimo. Si las funcionalidades no producen errores que impidan su correcto funcionamiento y si las mejoras propuestas no son imprescindibles a corto plazo.
- Requisitos:
  - Disponer la plataforma instalada en un servidor web de producción con los requisitos requeridos por la misma accesible para ser probados.
  - **Comunicación con los usuarios de la plataforma para poder recoger su *feedback*.**

### 3.2.4. Requisitos de implantación

El sistema tiene dos tipos de entorno el entorno de desarrollo y el de producción:

**Entorno de desarrollo (balieke.garatzen.net):** servirá para realizar pruebas antes de pasar cualquier cambio realizado en el sistema web a producción. Deberá consistir en un servidor que alojará toda la solución. Además deberá de estar disponible para los usuarios para que puedan probar la herramienta en diferentes fases del desarrollo.

<b>Marca</b>	Intel
<b>Modelo</b>	Celeron / Atom
<b>Frecuencia</b>	1.20+ GHz
<b>Arquitectura</b>	64 bits
<b>NIC</b>	FastEthernet
<b>RAM</b>	2 GB
<b>Disco duro</b>	1 TB

**Backup FTP**                      100 GB

**Entorno de producción (baleike.com):** será el que aloje la solución final y el que usen los clientes de ésta. Estará formado por dos servidores en los cuales se ejecutarán el sistema gestor de base de datos y el sistema gestor de contenidos.

La plataforma en su conjunto se ejecuta sobre un hosting compartido pero flexible en Dreamhost (300MB RAM asegurados). Más exactamente es un VPS (Virtual Private Server). Este tipo de hosting permite cambiar las características y recursos de la máquina respecto a las necesidades de manera dinámica. Sus características actuales son según el proveedor:

Disk Storage	<a href="#"><u>Unlimited TB + 50GB Backups</u></a>
Monthly Bandwidth	Unlimited TB
RAM	<b>300 MB</b> – 4000 MB
Domains Hosted	Unlimited
Full Shell / SSH / FTP / SFTP	Unlimited
Users	Unlimited
E-Mail (POP/IMAP/Webmail)	Accounts Unlimited
MySQL 5 Databases	Unlimited w/ Shared Hosting Plan, MySQL-optimized VPS, or Dedicated Server
Operating System	Debian Linux

Los requisitos del sistema son los mismos que se especifican en el apartado 2.3 de este mismo documento.

## 4. Desarrollo

En el desarrollo se tratarán sobre todos las partes relacionadas con la modificación y desarrollo. Los módulos que han requerido integración no requieren más que una explicación general de como se integran los *plugins* en Wordpress.

### 4.1. Planificación de las actividades de desarrollo e integración de sistema

El desarrollo se ha realizado tomando en cuenta la planificación general del proyecto especificada en el diagrama Gantt.

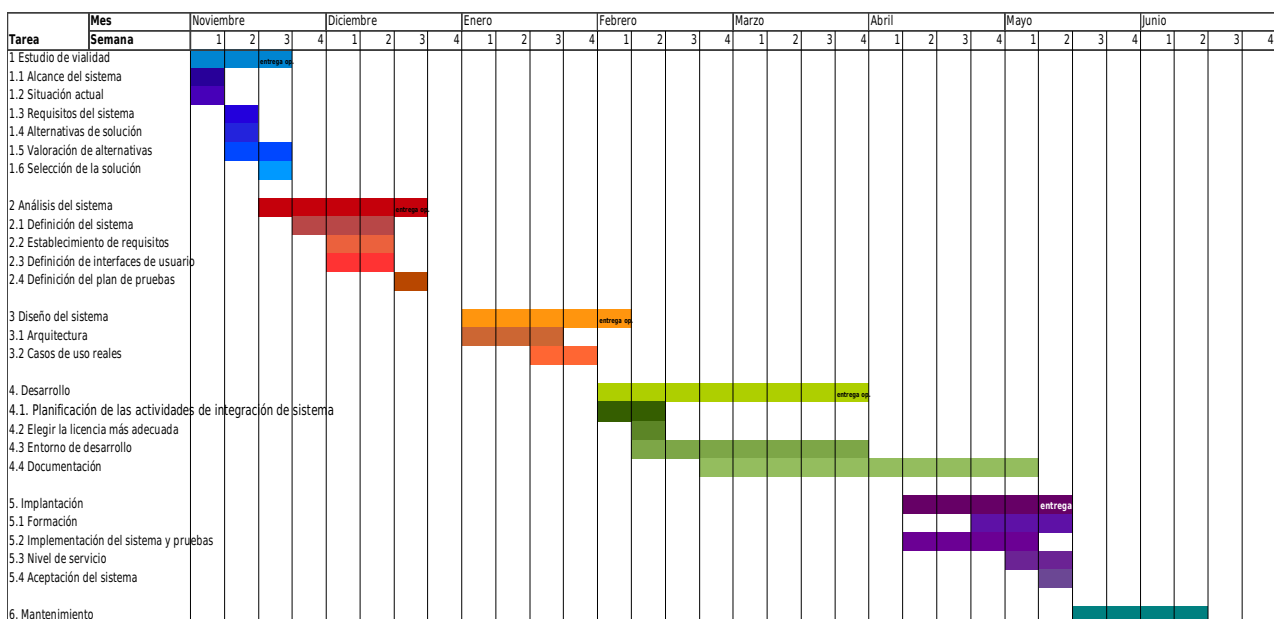


Ilustración 25: Diagrama Gantt de la planificación del proyecto

Dentro del mismo se han ido realizando en serie estas diferentes acciones.

- Preparación del entorno de desarrollo primero.
- Instalación y configuración y pruebas del CMS Wordpress.
- Integración, configuración y pruebas unitarias de los módulos que no requerían modificación.
- Integración, modificación, configuración y pruebas unitarias de los módulos que las requerían.

- Módulos que requerían de desarrollos más avanzados, integración, pruebas unitarias.
- Pruebas de integración.
- Pruebas de uso.

En paralelo se ha trabajado también el diseño de la aplicación. Este tema se tratará de forma colateral en el texto, haciendo referencia solamente a las decisiones técnicas interesantes que tienen que ver en el mismo (sin entrar a analizar el diseño en sí).

## 4.2. Desarrollo

En este apartado se analizarán las partes más importantes del desarrollo. Con ejemplos concretos de código en las ocasiones que así se requiera. Se diferencian cuatro apartados que tienen relación directa con los subsistemas y casos de usos de las fases anteriores.

### Instalación y configuración de Wordpress

La instalación y configuración de Wordpress ha seguido las pautas generales para una instalación normal de Wordpress ([http://codex.wordpress.org/Installing\\_WordPress](http://codex.wordpress.org/Installing_WordPress)).

Se ha configurado la plataforma en euskara modificando el fichero wp-config.php y añadiendo el *locale* eu.mo.

```
define('WPLANG', 'eu');
```

No es objeto de la memoria explicar la migración de los datos de la antigua red de blogs.

Simplemente contar que para ello se ha contado con un script de apoyo que se ha realizado a medida y la posterior revisión y comparación de las bases de datos.

Posteriormente y después de realizar la migración se han analizado todos los blogs de la plataforma de blog y se han tenido que realizar ciertos cambios en los mismos a mano. Ya que la compatibilidad entre versiones no era del todo perfecta.

Solucionado esto el desarrollo se centra en la web que representa el medio de comunicación baleike.com . Es a esta y no a las otras a las que no referimos a la hora de hablar del desarrollo de aquí en adelante.

### Integración de módulos genérica

La integración de módulos (*plugins*) en Wordpress se ha realizado de manera estándar. Probando de manera unitaria cada uno de ellos desactivando todos los demás para comprobar su correcto

funcionamiento.

Esto se ha llevado a cabo mediante el menú de administración de la plataforma y siguiendo los procedimientos de instalación común en estos casos

([http://codex.wordpress.org/Managing\\_Plugins](http://codex.wordpress.org/Managing_Plugins)) . Mediante la configuración específica de cada plugin se han configurado los mismos para cumplir las especificaciones del proyecto.

Plugina	Azalpena
<input type="checkbox"/> <b>Bad Behavior</b> Settings   Log   Whitelist   Ezgaitu	Deny automated spambots access to your PHP-based Web site. 2.2.6 bertsioa   Garatzailea: Michael Hampton   Bisitatu pluginaren gunea
<input type="checkbox"/> <b>Contact Form 7</b> Settings   Ezgaitu	Just another contact form plugin. Simple but flexible. 3.1.2 bertsioa   Garatzailea: Takayuki Miyoshi   Bisitatu pluginaren gunea
<input type="checkbox"/> <b>Feed Template Customize</b> Ezgaitu	This plugin modifies RSS feeds and ATOM feeds as you want. 1.0.0.1 bertsioa   Garatzailea: wokamoto   Bisitatu pluginaren gunea
<input type="checkbox"/> <b>kPicasa Gallery</b> Ezgaitu	Display your Picasa Web Galleries in a post or in a page. 0.2.9 bertsioa   Garatzailea: Guillaume Hébert   Bisitatu pluginaren gunea
<input type="checkbox"/> <b>Picasa Express x2</b> Settings   Ezgaitu	Browse, search and select photos from any publicly available Picasa Web Album and add them to your post/pages. 1.5.3 bertsioa   Garatzailea: Wott   Bisitatu pluginaren gunea
<input type="checkbox"/> <b>Post From Site</b> Ezgaitu	Add a new post/page/{your custom post type} directly from your website. 3.0.1 bertsioa   Garatzailea: Kelly Dwan   Bisitatu pluginaren gunea
<input type="checkbox"/> <b>Really Simple CAPTCHA</b> Ezgaitu	Really Simple CAPTCHA is a CAPTCHA module intended to be called from other plugins. It is originally created for my Contact Form 7 plugin. 1.5 bertsioa   Garatzailea: Takayuki Miyoshi   Bisitatu pluginaren gunea
<input type="checkbox"/> <b>Social Connect</b> Ezgaitu	Allow your visitors to comment, login and register with their Twitter, Facebook, Google, Yahoo or WordPress.com account. 0.9 bertsioa   Garatzailea: Brent Shepherd   Bisitatu pluginaren gunea

Ilustración 26: Visión de la administración relacionada con la gestión de los plugins

Algunos de ellos requerirán retoques estéticos desde el CSS pero sin necesidad de tocar el código más haya de su configuración.

## Módulos que han requerido modificaciones

De todos los *plugins* instalados algunos de ellos, referenciados ya antes en el apartado 3.2.2 han tenido que ser modificados. Esta es la relación concreta de los mismos y el resumen de los cambios realizados. Todos están documentados como se relata en el apartado 4.3. Esta es la relación de los *plugins* modificados.

### Post Featured Widget

Este bloque posibilita que noticias se coloquen de forma destacada en la columna central. Se modificar el modo de visualización para integrarlo con el formato de las noticias de la parte izquierda.

Los cambios realizados se encuentran en: */wp-content/plugins/post-feature-widget.2.1*

### **Post From Site**

Este *plugin* permite que directamente los usuarios puedan publicar desde la portada noticias. Para ello tuvimos que cambiar el rol por defecto de “suscriber” a “editor”.

Los cambios realizados se encuentran en: */wp-content/plugins/post-from-site*

### **Comment Indexer**

Este *plugin* fue modificado para integrarlo a la forma que querían los clientes de mostrar los comentarios de toda la plataforma en la portada.

Los cambios realizados se encuentran en: */wp-content/plugins/comment-indexer*

### **Wp NicEdit**

Este *plugin* que añade un editor avanzado en todas las *textareas* del sistema. Fue modificado para dejar sólo los botones que el cliente requería.

Los cambios realizados se encuentran en: */wp-content/plugins/wp-nicedit*

### **Social Connect**

Este *plugin* es el que permite identificarse utilizando (Twitter, Facebook, Google, Wordpress). Se modificó para cambiar el rol por defecto de los usuarios de “suscriber” a “editor” para que estos luego pudieran publicar desde portada utilizando las funcionalidades implementadas por el *plugin* Post from site.

Los cambios realizados se encuentran en: */wp-content/plugins/social-connect*

### **WPtouch Pro**

Este *plugin* permite la visualización por dispositivos móviles de nuestra web. Se modifica la visualización de la plantilla para estos dispositivos adaptándola a las necesidades del cliente.

Los cambios realizados se encuentran en: */wp-content/plugins/wptouch-pro*

## Desarrollos avanzados

### WP Issuu Viewer

Es sin duda el *plugin* que más cambios a necesitado. Con él se muestra una galería de documentos del servicio externo Issuu.com.

Debido a las especificaciones del cliente se tiene que hacer una galería, utilizando llamadas a la API de Issuu y con una interface que permita navegar entre los documentos.

Merece pararse en este proceso ya que utiliza ciertas funcionalidades de Wordpress que no son las comunes para generar una interface que se sale de la norma dentro de la plataforma.

Hemos tenido que primero modificar las funcionalidades del *plugin* para que realizará las correspondientes llamadas a la API además e hacer un pequeño cache en base a ficheros XML para que la carga no fuera demasiado grande.

Esto se puede verse concretamente en `/wp-content/plugins/wp-issuu-viewer/wpissuu.php` donde están comentados todos los cambios. En este mismo lugar se encuentra la carpeta cache en la que se recogen las llamadas a la API para no tener que repetirlas. Este es un ejemplo de un xml generado por el cache:

```
<?xml version="1.0" encoding="UTF-8"?>
<rsp stat="ok"><result totalCount="241" startIndex="0" pageSize="1"
more="true"><document documentId="120514191234-
0f5f22bafb384ef0bc3a3a79b5b3454f" title="209 (2012ko apirila) "
ep="1333576800" publishDate="2012-04-04T22:00:00.000Z" description="[Hitz-
tantak: Bixente Egaña] [Atez ate: Udal Gobernuaren eta EAJren iritziak]
[Iñigo Peña: Joko Olinpikoak dira nire ametsa] [Zelai: lan erreformaren
lehen biktimak]"/></result></rsp>
```

Para ello hemos tenido que crear una plantilla específica, siguiendo las instrucciones de la documentación ([http://codex.wordpress.org/Pages#Page\\_Templates](http://codex.wordpress.org/Pages#Page_Templates)). Nos hemos valido de esta funcionalidad por la necesidad de generar una interface completa dentro de Wordpress sólo para visualizar la galería de documentos de Issuu.

Este es el código de la plantilla `baleike_aldizkaria.php` que se puede encontrar dentro del *theme* `baleike_magacine` que luego trataremos.

```
<?php
/*
Template Name: Baleike aldizkaria
*/
```

```

?>
<?php get_header(); ?>
        <?php if (have_posts()) : while (have_posts()) : the_post();
?>
        <div class="post" id="post-<?php the_ID(); ?>">

                <h1><?php the_title(); ?></h1>

                <div class="entry">

                        <?php $subtitle = get_post_meta($post->ID,
'subtitle', true);

                                if($subtitle) echo '<p
class="sub">'.$subtitle.'</p>';

                                        ?>

                                <div class="argitalpenak_search_block">

                                        <div class="ezk">

                                                <h3>Baleike aldizkaria</h3>

                                                <a
href='/argitalpenak/baleike-aldizkaria/' class="button">Ikusi azken alea</a>

                                                        <form id="urteak" >

                                                                <label
for="selectidu">Urtea</label>

                                                                        <select
id="selectidu" onchange="top.location.href =
this.form.selectidu.options[this.form.selectidu.selectedIndex].value">

                                                                                <option
value="">-- Aukeratu urtea --</option>

                                                                                        <option
value="/argitalpenak/baleike-1994/">1994</option>

                                                                                                  <option
value="/argitalpenak/baleike-1995/">1995</option>

                                                                                                            <option
value="/argitalpenak/baleike-1996/">1996</option>

                                                                                                  <option
value="/argitalpenak/baleike-1997/">1997</option>

                                                                                                            <option
value="/argitalpenak/baleike-1998/">1998</option>

                                                                                                  <option
value="/argitalpenak/baleike-1999/">1999</option>

                                                                                                            <option
value="/argitalpenak/baleike-2000/">2000</option>

                                                                                                  <option

```



```

value="/argitalpenak/baleike-2001/">2001</option>
</select>
</form>
</div>
<!-- end .ezk -->
<div class="esk">
<h3>Aukeratu
argitalpena</h3>
<ul>
<li><a title="Argitalpenak
â€“ Arrauna" href="/argitalpenak/arrauna/">Arrauna</a></li>
<li><a title="Argitalpenak
â€“ Bazterbide" href="/bazterbide/">Bazterbide</a> </li>
<li><a title="Argitalpenak
â€“ Euskara" href="/euskara/">Euskara</a></li>
<li><a title="Argitalpenak
â€“ Hauteskundeak" href="/hauteskundeak/">Hauteskundeak</a></li>
<li><a title="Argitalpenak
â€“ Olagarroa" href="/olagarroa/">Olagarroa</a></li>
<li><a title="Argitalpenak

```

```

&eacute;" Poplitean" href="/poplitean/">Poplitean</a></li>
                <li><a title="Argitalpenak
&eacute;" TravelOut" href="/travelout/">TravelOut</a></li>
                <li><a title="Argitalpenak
&eacute;" Urtekariak" href="/urtekariak/">Urtekariak</a></li>
                <li><a title="Argitalpenak
&eacute;" Zumaia eta zumaiarrak" href="/zumaia-eta-zumaiarrak/">Zumaia eta
zumaiarrak</a></li>
            </ul>
        </div>
        <!-- end .esk -->
        <div class="clear"></div>
    </div>
    <!-- end .argitalpenak_search_block-->
<div class="argitalpenak">
<div id="azala">
<h2>Azken zenbakia</h2>
<?php
$xmlfile = '/home/baleike/www.baleike.com/wp-content/plugins/wp-issuu-
viewer/cache/azkena.xml';
if (file_exists($xmlfile)) {
    $xml = simplexml_load_file($xmlfile);
    $lau=0;
    //Loop the xml
    foreach($xml->result->children() as $first){
        $azk= $first->attributes();
        foreach($first->attributes() as $a => $result){
            }
        }
        echo '
                <object
classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
style="width:170px;height:210px" id="7cc1af6f-cd6e-b558-7ce4-adb98009935c"
class="azken_zenbakia">
                <param name="movie"
value="http://static.issuu.com/webembed/viewers/style1/v2/IssuuReader.swf?
mode=mini&amp;viewMode=singlePage&amp;shareMenuEnabled=false&amp;printButton

```

```

Enabled=false&amp;shareButtonEnabled=false&amp;searchButtonEnabled=false&amp
;documentId='. $azk.' " />

                                <param
name="allowfullscreen" value="true"/>

                                <param
name="allowscriptaccess" value="always"/>

                                <param name="menu"
value="false"/>

                                <param name="wmode"
value="transparent"/>

                                <embed style="text-align:
left; width: 175px; height: 210px;" width="320" height="240"
type="application/x-shockwave-flash"
src="http://static.issuu.com/webembed/viewers/style1/v2/IssuuReader.swf"
allowfullscreen="true" allowscriptaccess="always" menu="false"
wmode="transparent"
flashvars="mode=mini&amp;viewMode=singlePage&amp;shareMenuEnabled=false&amp;
printButtonEnabled=false&amp;shareButtonEnabled=false&amp;searchButtonEnable
d=false&amp;documentId='. $azk.' " />

                                </object>

                                <p class="title"><strong>' .
$first->attributes()->{"title"} . '</strong></p>

                                <p class="desc">' . $first-
>attributes()->{"description"} . '</p>;

                                }

?>

</div>

<div id="menu">

<h2>Urte zerrenda</h2>

<ul>

    <li><a title="1994" href="/argitalpenak/baleike-aldizkaria/baleike-
1994/">1994</a></li>

    <li><a title="1995" href="/argitalpenak/baleike-aldizkaria/baleike-
1995/">1995</a></li>

    <li><a title="1996" href="/argitalpenak/baleike-aldizkaria/baleike-
1996/">1996</a></li>

    <li><a title="1997" href="/argitalpenak/baleike-aldizkaria/baleike-
1997/">1998</a></li>

    <li><a title="1998" href="/argitalpenak/baleike-aldizkaria/baleike-
1998/">1998</a></li>

```

```
<li><a title="1999" href="/argitalpenak/baleike-aldizkaria/baleike-1999/">1999</a></li>

<li><a title="2000" href="/argitalpenak/baleike-aldizkaria/baleike-2000/">2000</a></li>

<li><a title="2001" href="/argitalpenak/baleike-aldizkaria/baleike-2001/">2001</a></li>

<li><a title="2002" href="/argitalpenak/baleike-aldizkaria/baleike-2002/">2002</a></li>

<li><a title="2003" href="/argitalpenak/baleike-aldizkaria/baleike-2003/">2003</a></li>

<li><a title="2004" href="/argitalpenak/baleike-aldizkaria/baleike-2004/">2004</a></li>

<li><a title="2005" href="/argitalpenak/baleike-aldizkaria/baleike-2005/">2005</a></li>

<li><a title="2006" href="/argitalpenak/baleike-aldizkaria/baleike-2006/">2006</a></li>

<li><a title="2007" href="/argitalpenak/baleike-aldizkaria/baleike-2007/">2007</a></li>

<li><a title="2008" href="/argitalpenak/baleike-aldizkaria/baleike-2008/">2008</a></li>

<li><a title="2009" href="/argitalpenak/baleike-aldizkaria/baleike-2009/">2009</a></li>

<li><a title="2010" href="/argitalpenak/baleike-aldizkaria/baleike-2010/">2010</a></li>

<li><a title="2011" href="/argitalpenak/baleike-aldizkaria/baleike-2011/">2011</a></li>

<li><a title="2012" href="/argitalpenak/baleike-aldizkaria/baleike-2012/">2012</a></li>

</ul>

</div>

<div class="clear"></div>

</div>

</div>

<div>

</div>

<?php wp_link_pages(array('before' =>
'<p><strong>'.__( 'Pages', "magazine-basic").':</strong> ', 'after' =>
'</p>', 'next_or_number' => 'number')); ?>

</div>

<?php //comments_template(); ?>

<?php endwhile; endif; ?>

<?php get_footer(); ?>
```

Después de crear la plantilla tiene que ser asignada mediante el menú de administración de esta forma:

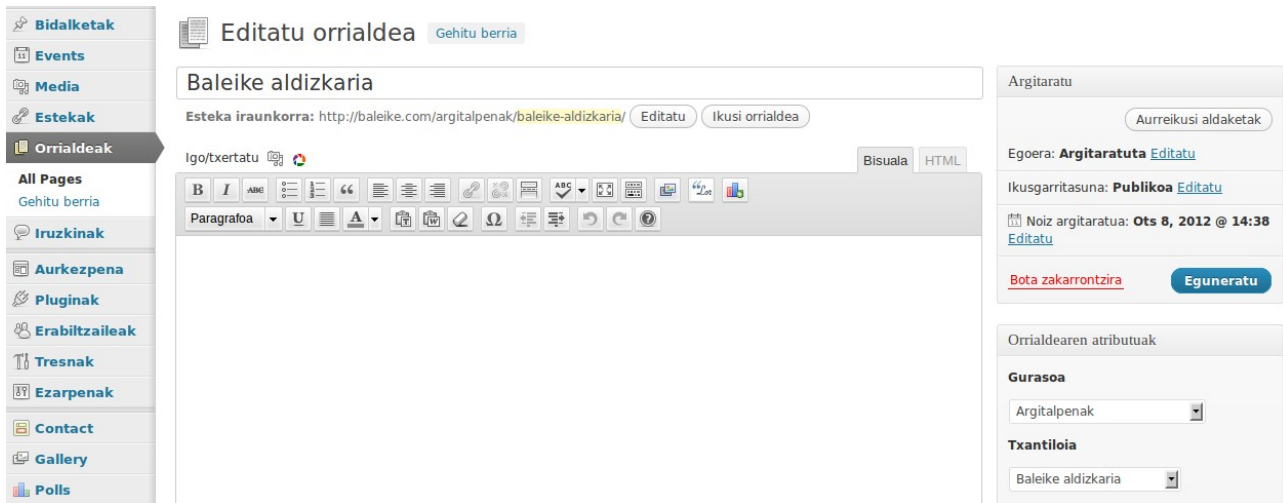


Ilustración 27: Visión de la administración relacionada con la asignación de la plantilla

De esta forma asignamos la plantilla concreta, a una página concreta, dentro de nuestra plataforma. El resultado se visualiza de esta manera.



Ilustración 28: Visión de la sección Argitalpenak (publicaciones)

Esto también ha requerido ciertos cambios en el diseño que se han integrado en el CSS de la plataforma.

Para que los nuevos números que se puedan subir al servicio Issuu vayan integrandose de forma

automática hemos configurado un *cron* para que haga las peticiones. De esta manera nos aseguramos que el *cache* este actualizado.

Además, de la misma manera el sistema de *cache* puede ser forzado si existe la necesidad de la incorporación inmediata de una documento a la galería de documentos.

## Theme

La plantilla es también interesante. Se ha usado para ello la formula del *child theme*. Esto es la plantilla está basada en otra. Se crea una carpeta nueva y se define mediante un fichero la relación entre ambas. A partir de este momento los ficheros que se introduzcan en la carpeta, y tengan el mismo nombre que los del padre, serán sobrescritos por estos.

Esto permite más modularidad, además de limpieza en el código. Ya que sólo se modifican los ficheros que cambian respecto a la plantilla padre. La carpeta que hemos creado se encuentra en `/wp-content/themes/baleike_magazine`. En el fichero `style.css` se define su plantilla padre:

```
/*
Theme Name: Baleike
Theme URI:
Description: Baleikeren tenplatea
Version: 1.0
Author: Elurnet @Erlantz
Author URI: http://www.elurnet.net/
Template: magazine-basic
Tags: three-column, grey
*/
```

Se han modificado también ciertos ficheros de los cuales el fichero `functions.php` es el más importante. En el se incluyen modificaciones utilizando los filtros y acciones de Wordpress ([http://codex.wordpress.org/Plugin\\_API#Hooks.2C\\_Actions\\_and\\_Filter](http://codex.wordpress.org/Plugin_API#Hooks.2C_Actions_and_Filter)) que modifican actitudes concretas en la plantilla.

```
<?php
register_sidebars( 1,
    array(
        'name' => 'goiesk',
        'before_widget' => '<div id="clicktopost">',
        'after_widget' => '</div>',
```

```

        'before_title' => '<h2>',
        'after_title' => '</h2>'
    )
);

register_sidebars( 1,
    array(
        'name' => 'oina',
        'before_widget' => '',
        'after_widget' => '',
        'before_title' => '<h2>',
        'after_title' => '</h2>'
    )
);

register_sidebars( 1,
    array(
        'name' => 'saresozialakezk',
        'before_widget' => '<div id="saresozialakezk">',
        'after_widget' => '</div>',
        'before_title' => '',
        'after_title' => ''
    )
);

register_sidebars( 1,
    array(
        'name' => 'saresozialakesk',
        'before_widget' => '<div id="saresozialakesk">',
        'after_widget' => '</div>',
        'before_title' => '',
        'after_title' => ''
    )
);

register_sidebars( 1,
    array(
        'name' => 'Publizitateaezk',
        'before_widget' => '<div id="publizitateaezk">',
        'after_widget' => '</div>',

```

```

        'before_title' => '',
        'after_title' => ''
    )
);
register_sidebars( 1,
    array(
        'name' => 'Publizitateask',
        'before_widget' => '<div id="publizitateask">',
        'after_widget' => '</div>',
        'before_title' => '',
        'after_title' => ''
    )
);
register_sidebars( 1,
    array(
        'name' => 'Publizitatea2',
        'before_widget' => '<div id="publizitatea2">',
        'after_widget' => '</div>',
        'before_title' => '',
        'after_title' => ''
    )
);

#####
function my_custom_login_logo() {
    echo '<style type="text/css">
        .login h1 a { background-image:url(/wp-
content/themes/baleike_magazine/images/logo-login.png) !important; }
    </style>';
}

add_action('login_head', 'my_custom_login_logo');

#####
function new_wp_trim_excerpt($text) {
    $raw_excerpt = $text;
    if ( '' == $text ) {

```



```

$text = get_the_content('');

$text = strip_shortcodes( $text );

$text = apply_filters('the_content', $text);
$text = str_replace(']]>', ']]>', $text);
$text = strip_tags($text,
'<object><param><embed><iframe><a><strong><em>');

$excerpt_length = apply_filters('excerpt_length', 55);

$excerpt_more = apply_filters('excerpt_more', ' ' .
'[...]');

$words = preg_split('/(<object.+</object>)|(<iframe.
+</iframe>)|(<strong.+</strong>)|\n|\r|\t|\s/', $text, $excerpt_length +
1, PREG_SPLIT_NO_EMPTY|PREG_SPLIT_DELIM_CAPTURE );

echo "<!--";
echo var_dump($words);
echo "-->";

if ( count($words) > $excerpt_length ) {
    array_pop($words);
    $text = implode(' ', $words);
    $text = $text . $excerpt_more;
} else {
    $text = implode(' ', $words);
}

}

return apply_filters('new_wp_trim_excerpt', $text, $raw_excerpt);
}

//remove_filter('get_the_excerpt', 'wp_trim_excerpt');
//add_filter('get_the_excerpt', 'new_wp_trim_excerpt');
/*
function nire_theme_excerpt(){
    $link = '<br /><a href="'.get_permalink().'" class="more-
link">'.__( "Read more &raquo;", "magazine-basic").'</a>';

    $text = get_the_content('');

```

```

$text = strip_shortcodes( $text );

$text = apply_filters('the_content', $text);
$text = str_replace(']]>', ']]>', $text);
$text = strip_tags($text, '<object><param><embed><iframe>');
$excerpt_length = apply_filters('excerpt_length', 55);

$excerpt_more = apply_filters('excerpt_more', ' ' .
'[...]');

$words = preg_split('/(<object.*?object>)|(<iframe.*?
iframe>)|\n|\r|\t|\s/', $text, $excerpt_length + 1, PREG_SPLIT_NO_EMPTY|
PREG_SPLIT_DELIM_CAPTURE );

if ( count($words) > $excerpt_length ) {
    array_pop($words);
    $text = implode(' ', $words);
    $text = $text . $excerpt_more;
} else {
    $text = implode(' ', $words);
}

echo "<p>".$text."</p><br /><a href='".$get_permalink()."'
class='more-link'>".__('Read more &raquo;', 'magazine-basic')."</a>";
}
*/

/* Twitter tools short url

add_filter( 'tweet_blog_post_url','modify_url' );

function modify_url ( $url ) {
    $url =
    return $url
}*/

// Admin barraren elementuak ezabatzeko Elurnet
function eliminar_nodos_admin_bar() {
    global $wp_admin_bar;

    // $wp_admin_bar->remove_menu('wp-logo'); // Elimina el logo de WordPress

```

```

(desaparece tambi n todo el submen )

$wp_admin_bar->remove_menu('about'); // Elimina el enlace "Sobre
WordPress"

$wp_admin_bar->remove_menu('wporg'); // Elimina el enlace a
wordpress.org

$wp_admin_bar->remove_menu('documentation'); // Elimina el enlace a la
documentaci n oficial (Codex)

$wp_admin_bar->remove_menu('support-forums'); // Elimina el enlace a los
foros de ayuda

$wp_admin_bar->remove_menu('feedback'); // Elimina el enlace
"Sugerencias"

//$wp_admin_bar->remove_menu('view-site'); // Elimina el submen  que
aparece al pasar el cursor sobre el nombre de la web

//$wp_admin_bar->remove_menu('comments'); // Elimina el acceso directo a
los comentarios

//$wp_admin_bar->remove_menu('updates'); // Elimina el icono de
notificaci n de actualizaciones

//$wp_admin_bar->remove_menu('new-content'); // Elimina el men  para
generar nuevo contenido

//$wp_admin_bar->remove_menu('my-account'); // Elimina el acceso a la
cuenta de usuario y al enlace para desconectarse
}

add_action('wp_before_admin_bar_render', 'eliminar_nodos_admin_bar');

// always show admin bar Elurnet
function pjw_login_adminbar( $wp_admin_bar) {
    if ( !is_user_logged_in() )
        $wp_admin_bar->add_menu( array( 'title' => __( 'Log In' ), 'href' =>
wp_login_url() ) );
}

add_action( 'admin_bar_menu', 'pjaw_login_adminbar' );
add_filter( 'show_admin_bar', '__return_true' , 1000 );

//hide wp-bannerize menu items to non administrators
if ( is_admin() && ! current_user_can( 'administrator' ) ) {
    add_action( 'admin_menu', 'mwr_remove_wpbannerize' );

    function mwr_remove_wpbannerize() {
        remove_menu_page( 'wp-bannerize-mainshow' );
    }
}

```

```

}

//hide wp-bannerize menu items to non administrators
if ( is_admin() && ! current_user_can( 'administrator' ) ) {
    add_action( 'admin_menu', 'mwr_remove_contact' );

    function mwr_remove_contact() {
        remove_menu_page( 'wpcf7' );
    }
}

//Elurnet

function email_suscribers($post_ID) {

global $wpdb;

$post = get_post($post_ID);

if ($post->post_date == $post->post_modified)
{
$title = "[Baleike] " . $post->post_title;

$message = "<p>" . $post->post_title . "</p>
<p>" . $post->post_content . "</p>
<p>Egin klik hemen osorik irakurtzeko: <a href='" .
get_permalink( $post_ID ) ."'>" . get_permalink( $post_ID ) . "</a></p>";

$headers = "From: baleike@baleike.com" . "\r\n";
$headers .= "MIME-Version:1.0\r\n";
$headers .= "Content-Type: text/html; charset=utf-8\r\nX-Mailer: PHP/" .
phpversion();

$email="baleike@googlegroups.com";
mail($email, $title, $message, $headers);
}

return $post_ID;

```

```
}  
  
add_action('publish_post', 'email_suscribers');  
  
//RSS katxea garbitu 30 minuturo  
add_filter( 'wp_feed_cache_transient_lifetime', create_function('$fixrss',  
'return 600;') );  
  
?>
```

En él se definen por ejemplo los espacios o *sidebars* ([http://codex.wordpress.org/Function\\_Reference/register\\_sidebars](http://codex.wordpress.org/Function_Reference/register_sidebars)) que podrán ser configurados utilizando *widgets* ([http://codex.wordpress.org/WordPress\\_Widgets](http://codex.wordpress.org/WordPress_Widgets)).

### 4.3 Documentación

El proceso ha sido documentado de diferentes maneras en el desarrollo:

- Durante el desarrollo se ha utilizado *git* para gestionar el código. Utilizando los comentarios en los *commits* se ha ido documentando las modificaciones y pasos correspondientes.
- En cada uno de los ficheros modificados se ha utilizado una sintaxis concreta mediante comentarios, para detectar de manera precisa donde se han realizado las modificaciones sin tener que acudir a otro lugar.
- Por su parte el proceso también se han generado algunos documentos de texto, incluidos en el fichero *tar.gz*.
  - *Baleike.com aldizkaria.odt* : detalles para el uso y configuración del *plugin wp-issuu-viewer* explicado extensamente anteriormente.
  - *Baleike crontab.txt*: configuración del *crontab*.
  - *Plugin eta txantiloi eguneraketak.odt*: Detalles de los *plugins* y *plantillas* utilizados durante todo el proceso de desarrollo.

## 5. Implantación y mantenimiento

La implantación se ha realizado según lo estimado y se ha llevado a cabo en dos fases. La fase que correspondía a los blogs que había que migrar y la implantación de la nueva web *baleike.com*.

## **5.1 Formación**

En el proceso de implantación se ha llevado a cabo una sesión presencial de formación sobre la plataforma de desarrollo con los usuarios finales (periodistas y administrador).

Para ello se ha requerido un curso de 5 horas de forma presencial, haciendo pruebas sobre la plataforma de desarrollo.

Esto ha servido para poder optimizar algunos flecos que habían quedado desde el punto de vista de la usabilidad de la plataforma. Las pruebas realizadas se detallan en el siguiente punto.

## **5.2 Implantación del sistema y pruebas**

La implantación se ha dado en dos fases:

1. Preparación del entorno de producción
2. Instalación y configuración y pruebas del CMS Wordpress sin módulos.
3. 1.fase: Migración de los blogs anteriores
  1. Bloqueo de la edición de los viejos blogs.
  2. Migración de la base de datos y el contenido.
  3. Análisis, solución de problemas y pruebas en blogs.
  4. Pruebas con usuarios.
4. 2. fase: Baleike.com
  1. Integración y pruebas unitarias de los módulos que no requerían modificación.
  2. Integración y pruebas unitarias de los módulos que han requerido modificaciones.
  3. Integración y pruebas unitarias de módulos que requerían de desarrollos más avanzados.
5. Pruebas de integración.
6. Pruebas de uso.

Las pruebas de uso se realizaron ya con los usuarios, en las que se distinguen también dos grupos:

1. grupo: Usuarios de los blogs
  1. Pruebas de detección de fallos de migración.
  2. Pruebas de publicación y gestión de contenido del blog.

3. Pruebas de uso de la administración del blog.
2. Grupo: Periodistas y administrador de baleike.com
  1. Pruebas de detección de fallos de migración.
  2. Pruebas funcionales de cada uno de los módulos integrados.
  3. Pruebas de identificación de la plataforma con servicios externos.
  4. Pruebas de utilización de servicios de recursos externos.
  5. Pruebas de interacción con servicios web externos (redes sociales).
  6. Pruebas de publicación y gestión de contenido de la plataforma.
  7. Pruebas de interacción (comentarios y envío de noticias) [junto a los usuarios de la plataforma].
  8. Pruebas de uso de la administración del blog.

### **5.3 Nivel de servicio**

Se propone al cliente un servicio de mantenimiento integral que este no acepta. Por lo que se ofrece un mantenimiento simple en base a problemas:

- Se estipulan la remuneración por hora trabajada y el tiempo de respuesta:
  - Problemas graves: 48 horas.
  - Problemas menores: una semana.
- Se estipula también la posibilidad de introducir mejoras para las cuales se remitirá a un presupuesto específico en cada caso.

### **5.4 Aceptación**

El proceso de aceptación pasa por mostrar el funcionamiento de la plataforma al cliente y a los usuarios finales. Nosotros utilizamos el tramo de formación para esto. El cliente en ese punto ya ha podido probar la plataforma en desarrollo durante mucho tiempo y en este preciso momento es cuando mostramos el funcionamiento y las pruebas realizadas.

Después del pase a producción y (repetición) de batería de pruebas internamente se acuerda con el cliente la aceptación formal. A partir de aquí empieza a correr un plazo de tres meses dónde pueden aparecer errores no detectados anteriormente a los que se toma el compromiso de responder.

## 6. Conclusiones

El proyecto resultante ha sido muy satisfactorio. Desde la consultoría, realizada hasta la implantación ha sido un proceso que sin duda ahora podemos decir que ha sido un éxito. Tanto en los objetivos, el nivel de aceptación por parte del cliente, el desarrollo y sobre todo por el proceso propio de aprendizaje que ha supuesto.

-----

La decisión de integrar en una sola plataforma (<http://baleike.com>) la red de blogs y el medio de comunicación (y no en dos Joomla y Wordpress) es sin duda uno de los grandes aciertos.

Demuestran, como decíamos que “el CMS Wordpress es ya más que una herramienta para crear blogs un framework de desarrollo”.

Empezando por el diseño en cuya labor de dirección he participado, es a mi parecer bastante interesante. Posiblemente no sea en lo meramente visual en donde se aprecie tanto; es sobre todo a nivel funcional y en facilidad dónde se han ganado enteros. Sobre todo después de mantener contacto con los periodistas el producto se ha conseguido adaptar a sus necesidades concretas.

El diseño no sólo hace referencia al diseño de la web, ya que desde un principio se han tenido en cuenta los dispositivos móviles. Por lo que tanto la red de blogs, como el medio de comunicación (la web principal) están pensados también para ser vistos desde un móvil.

Los objetivos funcionales definidos: noticias, noticias destacadas, noticias complementarias encuestas, eventos... Han sido conseguidos y ampliados en muchos casos. Se han adaptado al objetivo principal que consistía en “Diseñar de la web usando un CMS en software libre con un diseño adaptado al medio de comunicación”.

Otros dos aspectos importantes han sido también integrados en proyecto de manera satisfactoria: el uso de servicios web como recurso y la identificación mediante servicios web. Estos sin duda eran los objetivos más difíciles de alcanzar, pero también se han conseguido de una manera bastante integrada dentro de la misma.

La plataforma ahora cuenta con recursos como videos de YouTube, galerías de Picasa, galerías de documentos de Issuu. Y no sólo eso, también admite videos de otras plataformas como Vimeo de manera integrada además de dar la oportunidad de insertar en las columnas de código html no previsto.

Sin duda la posibilidad de gestionar de manera totalmente libre las columnas de la plataforma ha



sido sin duda una de las grandes mejoras. Ya que venían de una plataforma donde la personalización por parte del periodista normal era bastante tediosa.

A su vez, la integración del sistema de autenticación con, los sistema de Twitter, Facebook, Google, Wordpress hace que el acceso a la plataforma como usuario sea más fácil. Todo además, sin perder el control, ya que el usuario aunque se identifique utilizando una de esas plataformas crea un usuario local dentro de la plataforma.

Además de la identificación se ha logrado también integrar algunos de estos servicios en el sistema de una manera más práctica. Se posibilita el envío de cualquier noticia por parte de los usuarios a las redes sociales y el sistema está preparado para configurar los perfiles corporativos del medio y publicar automáticamente las noticias del medio en estos.

-----

El proyecto en su planificación a tenido una evolución correcta. Las estimaciones realizadas han sido correctas, aunque caso de fuerza mayor como la rotura de la clavícula del que les escribe ha generado un retraso de un mes en el proyecto. En ese periodo, una persona ha ido avanzado las partes más sencillas del desarrollo bajo mi supervisión.

De echar de menos algo, ha sido quizás la necesidad de haber detallado un poco más el análisis en concreto en la integración de Issuu en la plataforma. Aunque contábamos con ello, este ha sido sin duda uno de los puntos más difíciles en el proyecto. Teniendo que desarrollar casi desde cero un módulo entero.

Otras dificultades han tenido que ver con cierta falta de entendimiento con el cliente al principio. Para lo cual utilizamos herramientas como *wireframes* y como no, reuniones para detallar todo lo relacionado con el diseño y las funcionalidades del proyecto. Superados estos problemas iniciales el entendimiento fue inmejorable.

Sin duda de cambiar algo cambiaría la decisión de seguir utilizando un VPS como hosting. En nuestra labor de consultoría la mejor opción era la de utilizar un pequeño servidor dedicado. Esto no ha podido ser por motivos sobre todo económicos por parte del cliente.

-----

A futuro el proyecto podría ser un producto muy interesante para otros desarrollos. Como tal sería interesante paquetizarlo de alguna manera. Aunque, es difícil debido a la cadencia de actualizaciones de Wordpress, pero sin duda sería interesante.

Pueden surgir problemas con el Spam aunque se han interpuesto medidas para combatirlo. Creemos

que puede haber problemas relacionados con la publicidad no consentida al dar la posibilidad de publicación directa (aunque moderada) desde la portada. Quizás se debería de añadir algún sistema *captcha* también aquí.

Pero sin duda son dos las mejoras a futuro que se podrían realizar. Por una parte sería muy interesante integrar las redes sociales, sobre todo Facebook y Twitter no sólo para la identificación o difusión del contenido sino para la interacción. Permitir que la gente comente las noticias publicadas desde las mismas redes pero que tuviera reflejo también en la plataforma.

La segunda es quizás la posibilidad de generar no ya versiones móviles de las webs de la plataforma sino aplicaciones móviles para teléfonos Android y/o iOS. Estas dos son sin duda cosas a tomar en cuenta para un futuro.