



Rankitt:

Web social de encuestas

Memoria del proyecto de final de carrera correspondiente a los estudios de Ingeniería Superior en Informática presentado por Sheng Ping Wu Chen y dirigido por Javier Serrano García y Marc Bria.

Bellaterra, Septiembre de 2007

El sotasignante, Javier Serrano García, profesor de
l'Escola Tècnica Superior D'Enginyeria de la U.A.B,

CERTIFICA:

Que el trabajo al que corresponde la presente
memoria ha sido realizado bajo su dirección para
Sheng Ping Wu Chen.

Bellaterra, Septiembre de 2007

Signado: Javier Serrano García

Resum

Aquest document presenta una visió global del procés de desenvolupament de la pàgina web “rankitt.com”, una web social per a enquestes.

L'autor d'aquestes línies exposa els fonaments teòrics en els que es recolça el projecte i realitza l'anàlisi i diseny general de l'aplicació.

Resumen

En este documento se presenta una visión global del proceso de desarrollo de la página web “rankitt.com”, una web social para encuestas.

El autor de estas líneas expone los fundamentos teóricos en los que se apoya el proyecto y realiza el análisis y diseño general de la aplicación.

Abstract

This document is an overall view about the development process of a 3D Virtual Community of Learners prototype called “Shere Rom Space” (SRS).

The author explains project's theoretical main concepts and makes the application general analysis and design.

Índice

1. INTRODUCCIÓN	9
1.1. Definición encuesta	9
1.2. Tipos de encuesta	9
1.3. Rankitt: web social de Encuestas	10
1.4. Objetivos del proyecto	11
1.5. Estructura de memoria	11
2. HERRAMIENTAS Y ENTORNO DE DESARROLLO	12
2.1. Web 2.0	12
2.2. CMS	14
2.3. Drupal	17
2.4. Herramientas de desarrollo	17
2.4.1 Etapa de análisis	20
2.4.2 Etapa de diseño	20
2.4.3 Etapa de implementación	20
2.4.4 Etapa de pruebas	23
3. EL PROCESO DE DESARROLLO DE SOFTWARE Y METODOLOGÍA	24
3.1. El proceso de desarrollo de software	24
3.2. Metodología del Proceso Unificado	24
3.3. Aplicación de la metodología UP en el proyecto	29
4. ANÁLISIS	31
4.1. Definición de actores	31
4.2. Requerimientos funcionales	32
4.2.1 Diagramas de casos de uso	33
4.2.2 Especificación de los casos de Uso	34

4.3.	Requerimientos no funcionales	47
4.4.	Planificación	48
5.	DISEÑO Y IMPLEMENTACIÓN	50
5.1.	Diseño de datos	50
5.2.	Creación de tablas en Drupal	53
5.3.	Diseño de la interfaz	55
5.4.	Diagrama de pantallas	57
5.5.	Esquema de la arquitectura	59
5.6.	Diagramas de secuencias	60
5.7.	Creación de módulos en Drupal	64
6.	CONCLUSIONES	66
7.	BIBLIOGRAFIA	67
8.	GLOSARIO	68
9.	ÍNDICE FIGURAS	70
AI.	ANEXO I: DRUPAL	71

Agradecimiento:

A Marc y Daniel por haberme dado la oportunidad de realizar este proyecto.

Al profesor por los apoyos proporcionado.

A mi familia y a mis amigos.

1. Introducción

1.1 Definición de encuesta

*Una **encuesta** es un conjunto de cuestiones normalizadas dirigidas a una muestra representativa de la población o instituciones, con el fin de conocer estados de opinión o hechos específicos.*

- [Wikipedia](#) -

Conjunto de preguntas tipificadas dirigidas a una muestra representativa, para averiguar estados de opinión o diversas cuestiones de hecho.

- [Real Academia Española](#) -

1.2 Tipos de encuesta

Las encuestas tienen por objetivo obtener información estadística indefinida, mientras que los censos y registros vitales de población son de mayor alcance y extensión. Este tipo de estadísticas pocas veces otorga, en forma clara y precisa, la verdadera información que se requiere, de ahí que sea necesario realizar encuestas a esa población en estudio, para obtener los datos que se necesitan para un buen análisis. Este tipo de encuesta abarca generalmente el UNIVERSO de los individuos en cuestión.

Otro tipo de Encuestas es Encuestas por Muestreo en donde se elige una parte de la población que se estima representativa de la población total. Debe tener un diseño muestral, necesariamente debe tener un marco de donde extraerla y ese marco lo constituye el censo de población. La encuesta (muestra o total), es una investigación estadística en que la información se obtiene de una parte representativa de las unidades de información o de todas las unidades seleccionadas que componen el universo a investigar. La información se obtiene tal como se necesita para fines estadístico-demográficos.

Una forma reducida de una encuesta por muestreo es un "sondeo de opinión", esta forma de encuesta es similar a un muestreo, pero se caracteriza porque la muestra de la población elegida no es suficiente para que los resultados puedan aportar un informe confiable. Se utiliza solo para recolectar algunos datos sobre lo que piensa un número de individuos de un determinado grupo sobre un determinado tema.

1.3 Rankitt: Web Social de Encuestas

A medida que pasa el tiempo las tecnologías van evolucionando y junto a estos surgen nuevas necesidades. Gracias a Internet tenemos a nuestra disposición millones de personas como masa crítica para obtener estadísticas sobre sus necesidades, opiniones e inquietudes.

Para realizar una encuesta, éstas se realizaban en las calles o en las casas pero siempre en contacto de persona a persona. Este método se sigue aplicando pero fue sustituyendo poco a poco por las encuestas por teléfono o por las revistas. Estas son lentas, caras y a veces incorrectas, siempre depende de la gente que se escogió en su momento para realizar la encuesta. Por ejemplo si se realiza una encuesta sobre una masa crítica de 1000 personas se obtiene un 3% de error de muestreo. Si nos vamos a 10000 personas el error se reduce a 1%. Por lo tanto, cuantas más personas tomamos se tendrá unas estadísticas más fiables. Actualmente esto nos lo permite Internet.

Rankitt fue un proyecto propuesto por Daniel Martínez, licenciado en ADE. La idea principal de Rankitt es ser una web donde la gente pueda votar, no solamente con votar y ver el resultado, como lo hace la mayoría de web que proporcionan encuestas. Sino dejamos dar un paso más, para que la gente pueda opinar de la encuesta y sus votar sobre los comentarios que se generen.

Como cualquier web social hay que fomentar la participación de la gente y que ellos generen contenido para la web. Por lo tanto se pretende dar libertad para crear nuevas encuestas. Además premiaremos a las encuestas, que más participación tenga, situándola en la página principal.

Con la acumulación de datos, se estudiaría un método para hacer negocio. Se propuso en principio de crear unos API's para que los webs externos puedan compartir las encuestas en sus propias páginas web, un buen ejemplo son los videos online. Se dejó de un lado este tema debido a la complejidad que llegaría a ser para un proyecto fin de carrera. Aunque no se descarta como una propuesta de ampliación.

La web debe de tener una interfaz simple, organizado y vistoso para que el usuario pueda manejar fácilmente. Intentando ajustar a la definición de la usabilidad.

“la usabilidad es el rango en el cual un producto puede ser usado por unos usuarios específicos para alcanzar ciertas metas especificadas con efectividad, eficiencia y satisfacción en un contexto de uso especificado”

-definición norma ISO 9241-

1.4 Objetivos del proyecto

Los objetivos principales del proyecto son:

- Definir todas las funcionalidades necesarias para la realización de encuestas: proponer nuevas encuestas, votar en las encuestas, enviar a un amigo la encuesta.
- Web social donde la gente se reúna para debatir las encuestas.
- Permitir a los usuarios decidir el contenido a presentar con sus votos.
- Implementar un sistema de karma que se obtiene de la suma de puntos positivos y negativos que ha obtenido el usuario en sus comentarios.
- Definir un modelo de datos que permita guardar y gestionar toda la información necesaria.

1.5 Estructura de la memoria

El contenido de la memoria ha sido estructura de la siguiente forma, en el segundo capítulo hablaremos de las herramientas y entorno de desarrollo utilizado para abordar este proyecto. En este apartado daremos una pequeña explicación de que entendemos por Web 2.0 y explicaremos las herramientas que han sido utilizados en las diferentes fases de la realización del proyecto. Haremos un especial énfasis sobre el gestor de contenido Drupal. Que fue al final el CMS (content management system) que elegimos para abordar este proyecto.

En el capítulo 3 se va a introducir la metodología que se ha escogido para abordar este proyecto. Primero de todo introduciremos su descripción, luego como la metodología se ajusta a nuestro proyecto.

En el capítulo relacionado con el análisis describiremos que hacer en detalle y cuales problemas nos hemos encontrado a la hora de pensar como afrontarlos. Obtendremos los requerimientos funcionales y no funcionales del proyecto. Juntamente como una planificación temporal de las tareas en las que se ha dividido.

En el siguiente capítulo mostraremos de qué manera hemos resuelto los problemas planteados al proyecto. Veremos las interfaces que se han creado, las nuevas tablas que se ha tenido que crear aparte de las que veían con Drupal. Y detallaremos todo el proceso de implementación de los problemas que nos propusimos resolver.

Finalizaremos con las conclusiones que se han llegado después de finalizar el proyecto. Valoraremos si se ha conseguido los objetivos marcados.

2. Herramientas y entorno de desarrollo

Introduciremos unos conceptos para poder seguir la lectura de este proyecto.

2.1 Web 2.0

El término Web 2.0 fue acuñado por O'Reilly Media en 2004 para referirse a una segunda generación de Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs, los wikis o las folksonomias, que fomentan la colaboración y intercambio ágil de información entre los usuarios.

- [Wikipedia](#) -

En Web 1.0, un pequeño número de escritores creaban páginas web para un gran número de lectores. Como resultado, la gente podía obtener información yendo directamente a la fuente. Con el tiempo, sin embargo, más y más personas comenzaron a escribir contenido además de leerlo. Esto tuvo efectos interesantes, repentinamente había demasiada información como para seguirle el paso. No teníamos suficiente tiempo para visitar todos los sitios con contenidos relevantes. A medida que la edición personal se popularizó y se convertía en la corriente dominante, se hizo evidente que el paradigma web 1.0 tenía que cambiar.

La Web 2.0 es la clara transición que se tenía que dar hacia la creación de aplicaciones web enfocadas al usuario final. Además de proporcionar servicios para reemplazar las aplicaciones de escritorio. La Web 2.0 es una actitud y no precisamente una tecnología.

The Web of documents has morphed into a Web of data. We are no longer just looking to the same old sources for information. Now we're looking to a new set of tools to aggregate and remix microcontent in new and useful ways.

- [Joshua Porter and Richard MacManus](#) -

Según Joshua y Richard, ahora la información es desmenuzada en unidades de microcontenidos para poder ser distribuidas a través de Internet y se intenta obtener nuevas herramientas para agregar y trabajar con esos microcontenidos. Estas herramientas denominadas interfaces Web 2.0, ya están presente en nuestro alrededor, solamente hay que detenerse y verlas: agregadores RSS, motores de búsqueda, portales, APIs (Interfaz de programación de la aplicación, la cual provee conexiones a los datos) y servicios Web. Las herramientas como estas están cambiando la manera que almacenamos y accedemos sin importar donde este situado el contenido.

Durante la Web Conference del 2005 se resumieron los principios clave que creen que caracterizan a las aplicaciones Web 2.0:

- La web es la plataforma.

- La información es el procesador.
- Efectos de la red movidos por una arquitectura de participación.
- La innovación surge de características distribuidas por desarrolladores independientes.
- El fin del círculo de adopción de software (“Servicios en beta perpetuo”).
- El software por encima de un solo aparato.

Para ello crearon un mapa meme en el cual coloca la web como plataforma donde no tiene una clara frontera, sino más bien, un núcleo gravitacional. Se puede visualizar Web 2.0 como un sistema de principios y practicas que conforman un verdadero sistema solar de sitios que muestran algunos o todos esos principios.

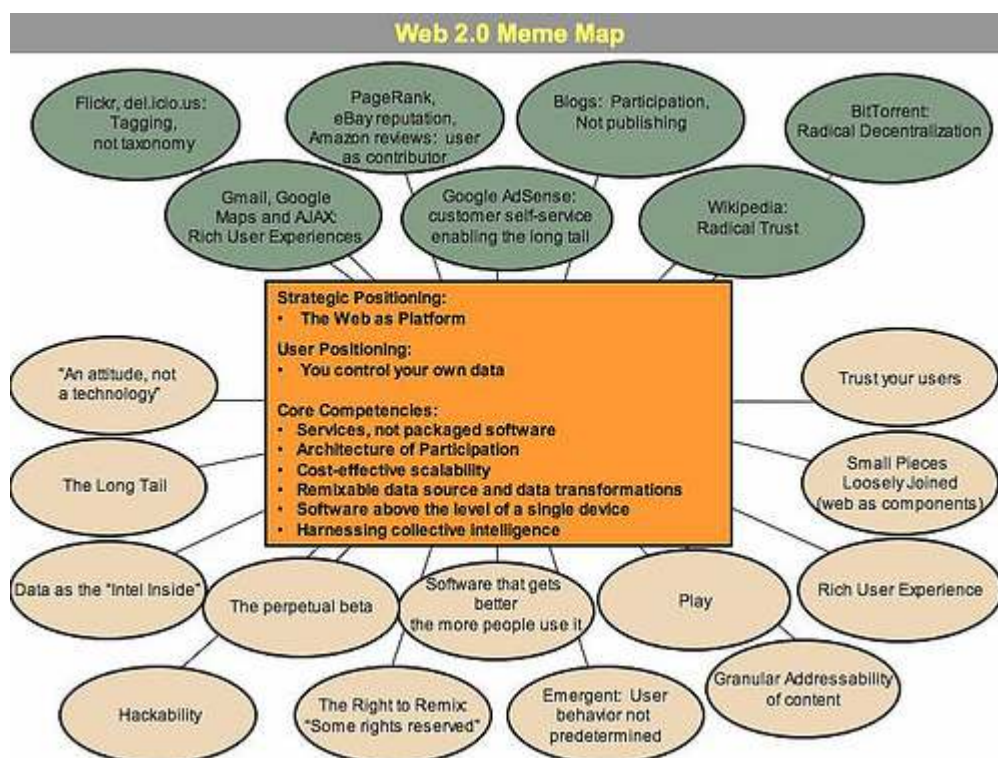


Figura: 2-1 “Mapa meme” de la web 2.0 realizada en una conferencia en O’Reilly Media

Una web se puede decir que esta construida usando tecnología de la Web 2.0 si se caracteriza por las siguientes técnicas:

- Transformar software de escritorio hacia la plataforma de la web.
- Respecto a los estándares del XHTML.
- Separación de contenido del diseño con uso de hojas de estilo (CSS).
- Sindicación de contenidos (RSS, RDF, Atom).
- Técnicas de aplicaciones ricas no intrusitas (como AJAX).
- Uso de Flash, Flex o Lazlo.
- Uso de Ruby on Rails para programar paginas dinámicas.
- Utilización de redes sociales al manejar usuarios y comunidades.
- Dar control total a los usuarios en el manejo de su información.

- Proveer APIs o XML para que las aplicaciones puedan ser manipulados por otros.
- Facilitar el posicionamiento con URL sencillos.

A continuación les mostramos una comparación de Web 1.0 y sus replicas en Web 2.0

Web 1.0	Web 2.0
DoubleClick	Google AdSense
Ofoto	Flickr
Akamai	BiTorrent
Mp3.com	Napster
Enciclopedia Británica	Wikipedia
Webs Personales	Blogging
Evite	Upcoming.org
Directorios (Taxonomía)	Etiquetas (Folksonomía)
Stickiness	Sindicación

2.2 CMS

Un Sistema de gestión de contenidos (Content Management System, en inglés, abreviado CMS) permite la creación y administración de contenidos principalmente en páginas web.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público.



Figura: 2-2 Esquema de un CMS.

El término Content Management System fue originalmente usado para la publicación de sitios web. Los primeros sistemas de administración de contenidos fueron desarrollados internamente por organizaciones que publicaban mucho en internet, como revistas en línea, periódicos y publicaciones corporativas. En 1995, el sitio de noticias tecnológicas CNET sacó su sistema de administración de documentos y publicación y creó una compañía llamada Vignette, que abrió el mercado para los sistemas de administración de contenido comerciales.

Conforme el mercado ha evolucionado los productos marcados como CMS han ampliado el panorama fragmentando el significado del término. Los wikis y los sistemas groupware también son incluidos como CMS.

Un sistema de administración de contenido a menudo funciona en el servidor del sitio web. Muchos sistemas proporcionan diferentes niveles de acceso dependiendo el usuario, variando si es el administrador, editor, o creador de contenido. El acceso al CMS es generalmente vía el navegador, y a veces se requiere el uso de FTP para subir contenido, generalmente fotografías o audio.

Los creadores de contenido crean sus documentos en el sistema. Los editores comentan, aceptan o rechazan los documentos. El editor en jefe es responsable por publicar el trabajo en el sitio. El CMS controla y ayuda a manejar cada paso de este proceso, incluyendo las labores técnicas de publicar los documentos a uno o más sitios. En muchos sitios con CMS una sola persona hace el papel de creador y editor, los blogs generalmente funcionan de esta manera.

La multitud de diferentes CMS existentes se pueden agrupar en las siguientes categorías:

- Foros: sitio que permite la discusión en línea donde los usuarios pueden reunirse y discutir temas en los que están interesados.
- Blogs: Publicación de noticias o artículos en orden cronológico con espacio para comentarios y discusión.
- Wikis: Sitio web dónde todos los usuarios pueden colaborar en los artículos, aportando información o reescribiéndola. También permite espacio para discusiones. Indicado para material que irá evolucionando con el tiempo.
- eCommerce: Son Sitios web para comercio electrónico.
- Sitios web: Sitio web con contenido y funcionalidad diversa que sirve como fuente de información o como soporte a una comunidad.
- Galería: Permite administrar y generar automáticamente un portal o sitio web que muestra contenido audiovisual, normalmente imágenes.

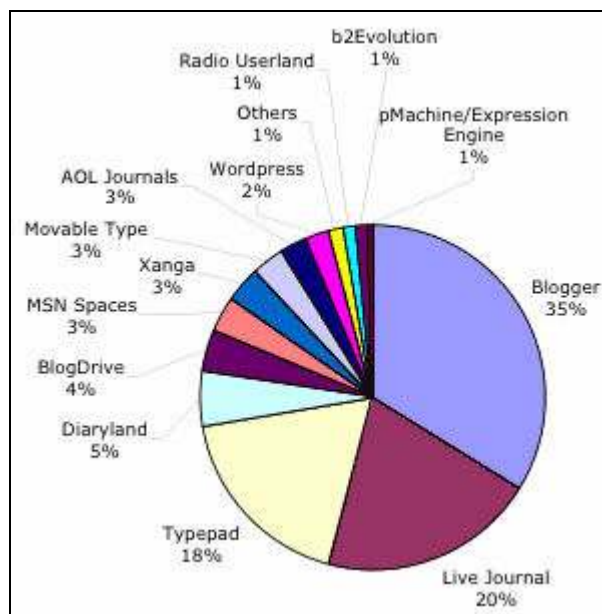


Figura: 2-3 El mercado de los CMS para los bloggers según Google.

2.3 Drupal

Drupal es un sistema de gestión de contenidos (CMS) muy completo y flexible creado en sus inicios por Dries Buytaert. Está construido con PHP y es software libre. Tiene, por todo esto, una gran comunidad alrededor de él mejorando su código y creando módulos que lo amplían y lo flexibilizan aún más. Con esto queda claro que muchos sitios web de la red lo utilizan de manera profesional.

Si no estás familiarizado con Drupal, solo tienes que imaginarlo como una montaña gigante de piezas de Lego (o Tente) para construir cualquier tipo de sitio web. Drupal es una plataforma increíblemente potente. Su sistema modular y framework básico permiten desarrollos rápidos de sitios con muchas funcionalidades. Todo lo que necesitas es imaginar que quieres construir y empezar a poner una pieza sobre otra.

- Jeff Robbins, diseñador y cofundador de Lullabot -

El gran sistema de permisos de usuario de Drupal ha hecho que se le conozca como una plataforma de creación de comunidades web y su APO permite a los plugins alterar los datos, crear páginas y contenidos y cambiar la manera de la que el sitio trabaja. Además del gestor de contenido básico para blogs, su flexible sistema ha permitido crear módulos para gestionar imagen, audio, vídeo, añadir información de tiempo y fecha al contenido y capacidad para mostrarlo en un calendario y/o en formato iCal o RSS, etiquetar cualquier contenido con coordenadas geográficas y colocarlo en un mapa o permitir buscarlo geográficamente.

Hay módulos que añaden funcionalidades de comercio electrónico, gestionar listas de contactos o que permiten mensajes privados entre usuarios así como establecer grupos privados o públicos a lo largo de todo el sitio donde los usuarios pueden crear sus propios grupos de interés con listas de correo, calendarios, etc. Incluidos. Y muchísimas más de esas piezas de construcción (por encima de 700 ahora mismo) están siendo proporcionadas por una comunidad internacional de programadores todos los días.

En el anexo I hemos dado una explicación más extensa sobre funcionamiento de Drupal.

2.4 Herramientas de desarrollo

Durante el desarrollo de este proyecto se ha intentado utilizar herramientas con licencia GPL (GNU Licencia Publica General). Menos algunas fases del proyecto, hemos tenido que utilizar Macromedia Flash para mostrar las graficas de las encuestas. La plataforma base a utilizar ha sido Drupal.

En su inicio exploramos varias alternativas: Joomla, Plone. Junto con Drupal son los tres sistemas Open source CMS más famoso y utilizado. Al Open source nos referimos que no es necesario tener una licencia para utilizarla.

En el siguiente apartado veremos una breve comparación entre los tres sistemas. Y al final argumentaremos porque hemos elegido a Drupal como nuestra plataforma base.

Comparativa entre Joomla, Drupal y Plone:

Que mejor manera de medir la popularidad observando el número de búsqueda en el buscador google. Si observamos en la siguiente gráfica, se observa claramente que Joomla gana con gran diferencia en número de búsqueda. Eso repercute su popularidad que aumenta respecto a los restos de CMS. En el otro lado, Drupal se sitúa en el segundo lugar gracias a su aumento de popularidad en los últimos años, donde ha desbancado a xoops de la segunda posición.

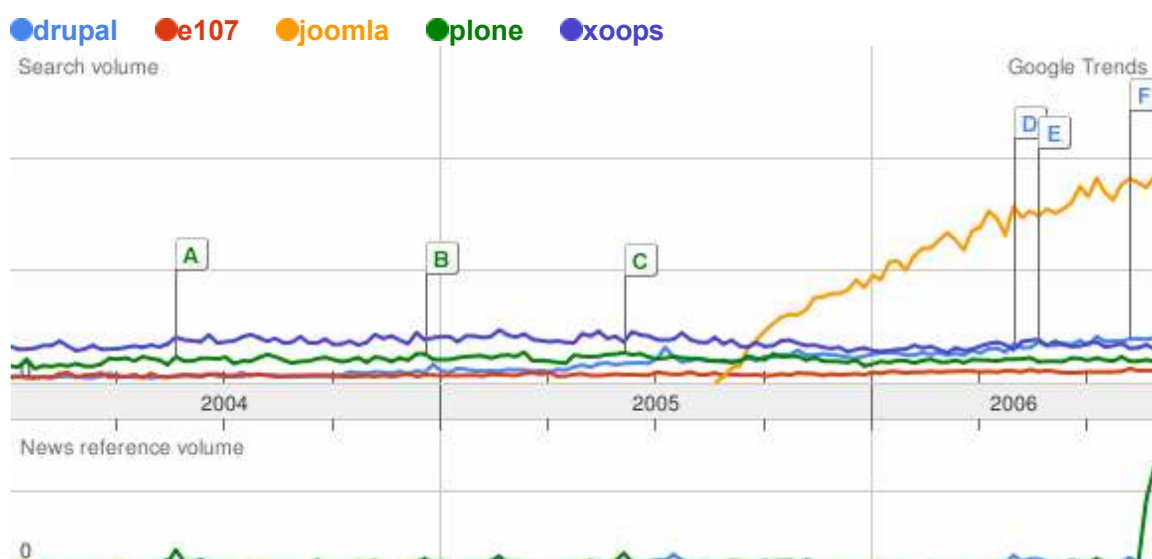


Figura: 2-4 Números de búsquedas realizadas sobre los distintos CMS.

Pero si observamos a la cantidad de visita que recibe las web donde se lleva a cabo los proyectos, las diferencias no se aprecian demasiado claro en este caso entre los dos CMS más buscados. Los dos tienen un comportamiento muy paralelo. Esto nos viene a confirmar que estos dos CMS son los más populares actualmente en volumen de búsqueda y en volumen de accesos.

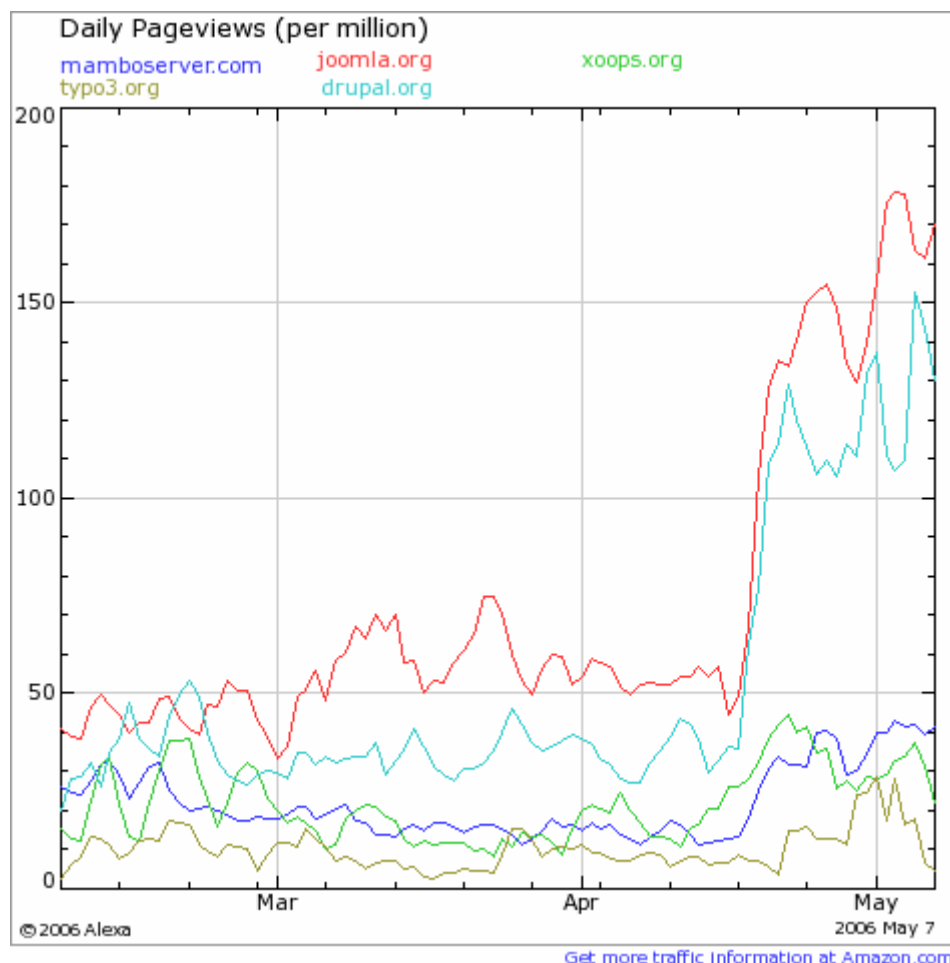


Figura: 2-5 Cantidad de acceso a las páginas a los webs del proyecto.

	Drupal	Mambo	Typo3	Movable type	Word press	Text pattern
Ease of install	●	●	○	●	●	●
Learning curve	●	●	○	●	○	○
Session control	●	●	●	○	○	○
User control	●	●	●	●	●	●
Extensability	●	●	●	●	○	○
Scalability	●	●	●	●	○	○
Themability	●	●	●	●	○	○
xHTML/CSS	●	●	○	●	●	●

Figura: 2-6 Propiedades de los distintos sistemas.

Según la tabla anterior el sistema Drupal es un poco difícil de instalar, ya que no posee en la última versión (4.7) un instalador por defecto y hay que realizarlo manualmente. Pero en tema de escalabilidad y de crear nuevos temas (Themability) son dos puntos a favor hacia Drupal. Porque pensamos que a la larga rankitt tendría que soportar a más de 100.000 usuarios. Por eso

la escalabilidad es muy importante para que el sistema responda correctamente sin ralentizar la navegación. En el “Themability” deseamos un sistema como Drupal que sea flexible para poder adaptarnos nosotros nuestro propio tema y no los que vienen por defecto. Además Drupal al estar organizado en módulos la curva de aprendizaje es más corto, porque las cosas están mejor estructurado. En la figura se observa que aparece Mambo y no Joomla. Es debido que Joomla es el nombre actual del proyecto inicial Mambo.

2.4.1 Etapa de análisis

Para la etapa de análisis hemos utilizado las siguientes aplicaciones:

GRANTTPROJECT

Esta aplicación es un planificador de proyecto muy sencillo, como su nombre indica nos sirve para realizar diagramas de Grant. Hemos hecho servir este programa para crear la planificación de tareas del proyecto y su durada.

Drupal

Durante esta etapa hemos tenido que documentarnos sobre el funcionamiento de este CMS.

2.4.2 Etapa de diseño

Para crear las distintas diagramas he utilizado el programa **DIA** que permite generar diagramas vectoriales.

Para retocar las imágenes se ha utilizado una versión trial de Photoshop de 30 días.

2.4.3 Etapa de implementación

Para la programación de los nuevos módulos hemos tenido que utilizar el lenguaje de programación PHP con algunas funciones en JavaScript y para el tema de estética el CSS y HTML.

HTML y CSS

HTML acrónimo de *HiperText Markup Lenguaje* es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

CSS son las siglas de Cascading Style Sheets. Las hojas de estilo en

cascada son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML.

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a una web entera, a un documento HTML, a una porción del documento o a una etiqueta en concreto. Con CSS conseguimos separar la capa de código del documento de su presentación lo que proporciona potencia a la programación.

PHP

PHP '*Hypertext Preprocessor*' es un lenguaje de programación interpretado de alto nivel, multiplataforma, cuyos programas se ejecutan en la parte del servidor, y cuyo código se escribe incrustado con el código HTML.

Dado que es multiplataforma funciona tanto para Unix, Windows y MAC OS, de forma que el código desarrollado se puede ejecutar en cualquiera de ellos sin modificaciones.

Al ejecutarse en el servidor, será necesario pues, que tengamos un servidor web para poder comprobar nuestros programas. La opción que se tomó en la generación del proyecto fue Apache.

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes para el usuario, al que le parecerá que está visitando páginas HTML que cualquier navegador puede interpretar.

PHP tiene muchas ventajas, entre ellas:

- Software libre, usa la licencia GPL.
- Muy sencillo de aprender. Similar en sintaxis a C y a Perl.
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- Toda la comprobación de la información enviada por el cliente se hace en el servidor.
- PHP proporciona unos tiempos de ejecución óptimos.

Principales usos del PHP:

- Programación de páginas web dinámicas en servidores, habitualmente en combinación con MySQL, aunque cuenta con un soporte para muchas otras bases de datos.
- Creación de aplicaciones gráficas 'stand-alone' que no usan navegador web.

JAVASCRIPT

JavaScript es un lenguaje de programación interpretado que permite la ejecución de código dentro de las páginas HTML con una sintaxis semejante a la del Lenguaje Java. A diferencia de PHP y al igual que HTML, todo el código JavaScript se ejecuta en el cliente.

Gracias a los programas escritos en este lenguaje y embebidos en las páginas HTML, se pueden conseguir interesantes efectos en las páginas web, como comprobar la validez de la entrada de formularios, abrir y cerrar ventanas, cambiar dinámicamente el aspecto y los contenidos de una página, cálculos matemáticos sencillos y mucho más sin necesidad de realizar ninguna transmisión de datos por la red.

En nuestra aplicación todo el código Javascript utilizado es completamente compatible en todos los navegadores.

MySQL

MySQL es un gestor de bases de datos de código abierto, lo que quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Podemos descargar MySQL desde Internet y usarlo sin ningún coste, de ese modo cualquiera puede inclinarse a estudiar el código fuente y modificarlo en base a sus necesidades. MySQL usa el GPL para definir qué podemos y no podemos hacer con el software en diferentes situaciones.

MySQL es muy rápido, fiable, robusto, multihebra, multiusuario, multiproceso, fácil de usar, con capacidad para manejar tanto grandes volúmenes de datos como pequeños, disponible para varias plataformas, acceso seguro mediante privilegios y password, soporta ODBC, basado en ANSI SQL 92, arquitectura cliente/servidor, etc.

MySQL es un Sistema de Gestión de Base de Datos Relacional. El modelo relacional se caracteriza, a muy grandes rasgos, por disponer de toda la información contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos. Esto añade velocidad y flexibilidad.

Su sencillez de instalación y su gran compatibilidad con lenguajes de programación de servidores web como PHP, hacen de esta herramienta el soporte perfecto a páginas web dinámicas. De hecho, en los últimos tiempos han surgido paquetes instaladores que montan, sobre cualquier plataforma, un servidor Apache, PHP y MySQL sin dificultad. En este proyecto se ha usado uno de estos paquetes, el paquete XAMPP.

Notepad ++

Editor gratuito de código fuente, que nos ha permitido editar los ficheros y programar nuevas funciones para Drupal.

Flash

Para solventar los problemas de incompatibilidad entre los navegadores Firefox y IE en los CSS. Se ha tenido que decidir en utilizar Flash para crear una película con actionscript para mostrar la gráfica y las bombillas.

Drupal

Durante esta fase hemos tenido que añadir nuevas funciones (módulos) para poder votar en las encuestas y en los comentarios. Y el cálculo total de las encuestas más votadas del día, de la semana y del mes.

2.4.4 Etapa de pruebas

Para esta última etapa hemos utilizado el paquete XAMPP junto con el navegador Firefox y Internet Explorer para comprobar que todo funcionaba a la perfección.

3. El proceso de desarrollo de software y metodología

3.1 El proceso de desarrollo de software

La Ingeniería del Software se podría definir como el establecimiento y aplicación de principios de la Ingeniería para obtener software. Teniendo en cuenta factores tan importantes como el coste económico, la fiabilidad del sistema y un funcionamiento eficiente que satisfaga las necesidades del usuario.

El proceso de desarrollo de software “es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo”. Concretamente “define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo”

- Jacobson -

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso se le llama el ciclo de vida del software que comprende cuatro grandes fases:

- **Concepción:** define el alcance del proyecto y desarrolla un caso de negocio.
- **Elaboración:** define un plan del proyecto, especifica las características y fundamenta la arquitectura.
- **Construcción:** crea el producto.
- **Transición:** transfiere el producto a los usuarios.

3.2 Metodología del Proceso Unificado

El Proceso Unificado (UP) es un método iterativo de desarrollo de software que guía a los equipos de proyecto en cómo administrar el desarrollo de un modo controlado mientras se equilibran los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto.

El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura.

Las características primordiales de la metodología del Proceso Unificado son:

- Iterativo e incremental:** el UP está compuesto de cuatro fases denominadas Concepción, Elaboración, Construcción y Transición. Cada una de estas fases se divide a su vez en una serie de iteraciones o ciclos (la de concepción sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. Cada una de estos ciclos se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada.

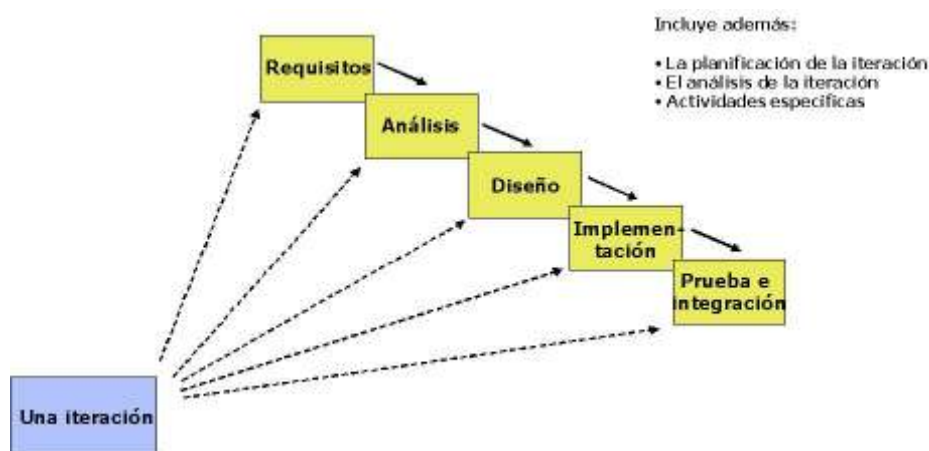


Figura: 3-1 Disciplinas de cada Iteración.

En la siguiente figura se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto UP.

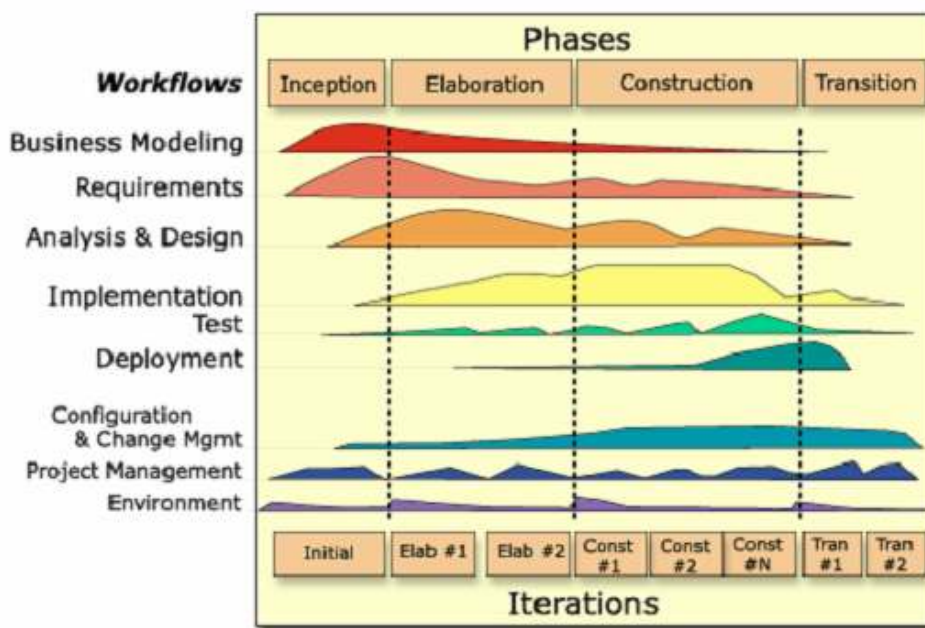


Figura: 3-2 Esfuerzo asociado a las disciplinas según la fase.

- **Centrado en la arquitectura:** la arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe de ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de UP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

- **Guiado por casos de uso:** en el UP los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que en cada iteración, basándose en los Casos de Uso, se creen los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso.

En la siguiente figura se muestra el camino que se sigue en las diferentes actividades del proceso de desarrollo a partir de los Casos de Uso.

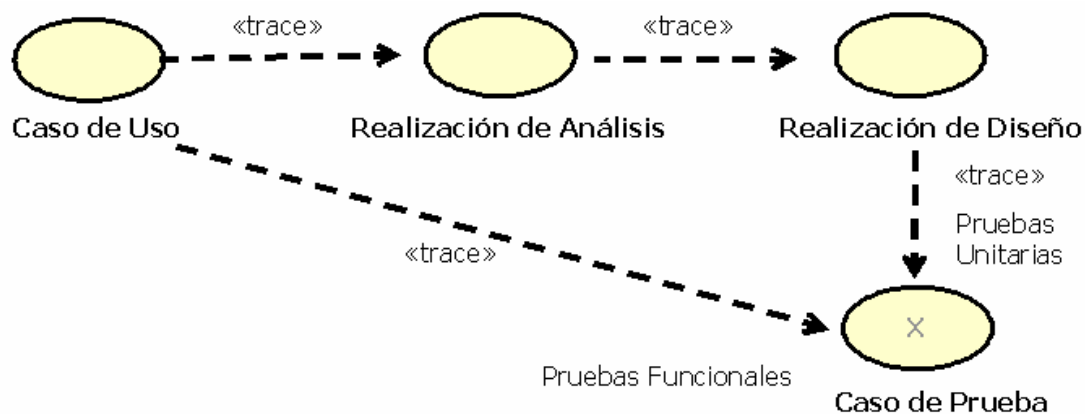


Figura: 3-3 Proceso de desarrollo a partir de los Casos de Uso.

- **Confrontación de riesgos:** el UP requiere que el equipo de proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

A continuación se muestra un esquema gráfico de Proceso Unificado de desarrollo de Software.

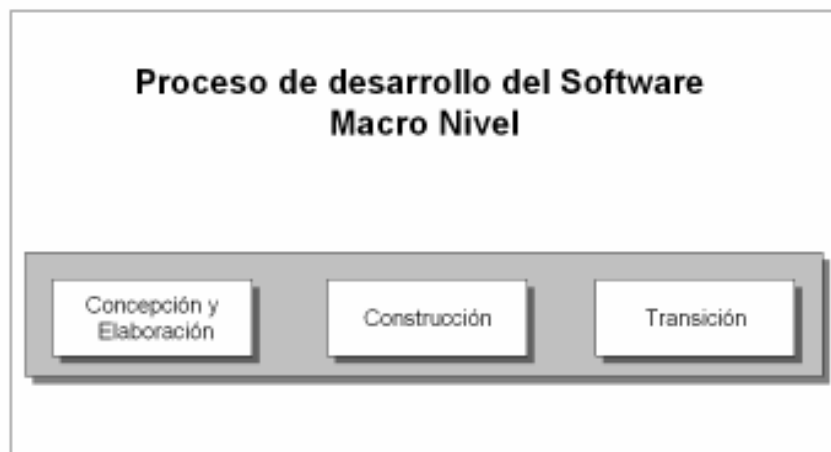


Figura: 3-4 Proceso de desarrollo de Software Macro Nivel.

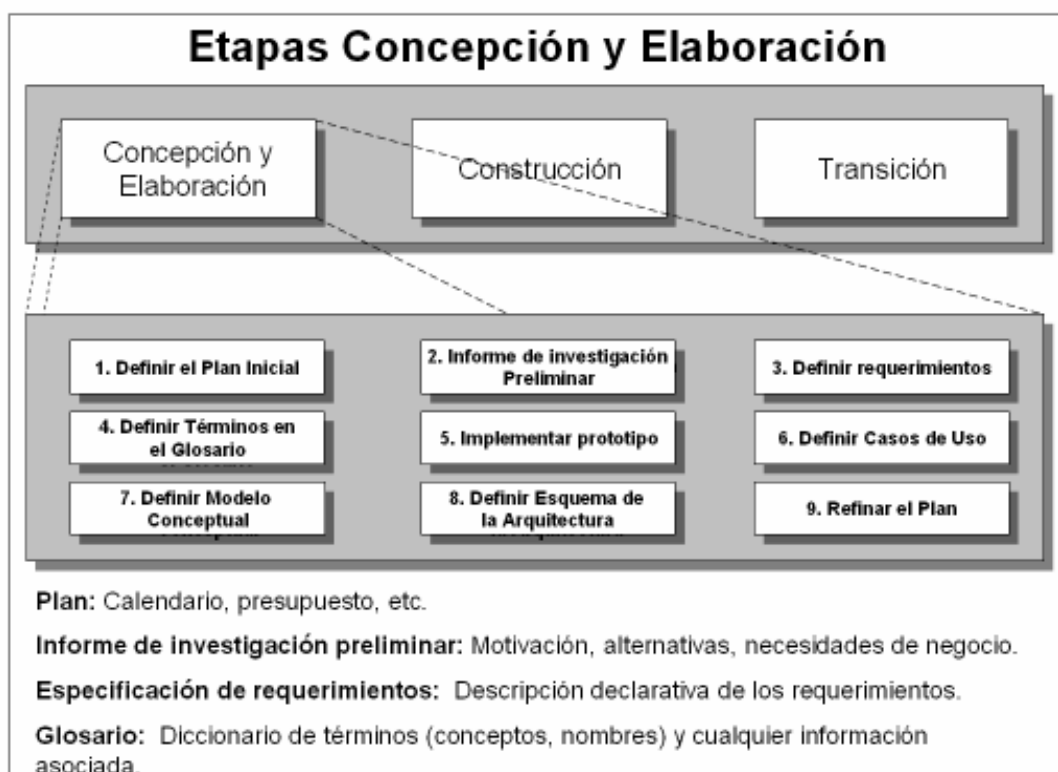


Figura: 3-5 Etapas Concepción y Elaboración.

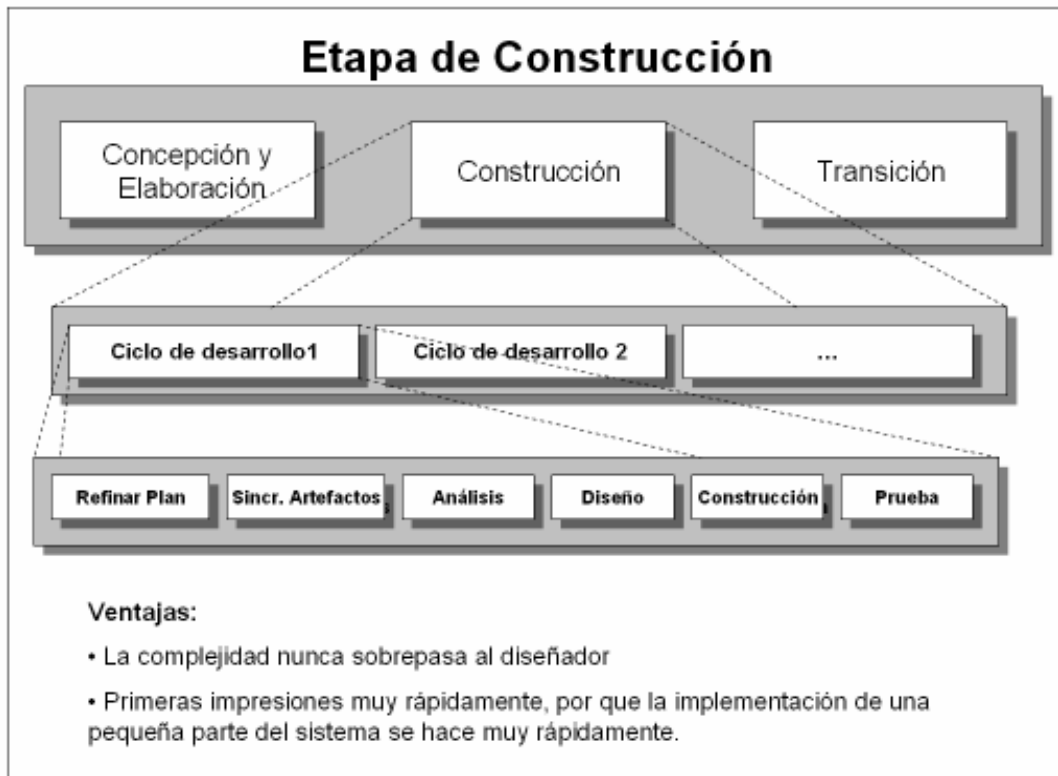


Figura: 3- 6 Etapa de Construcción.

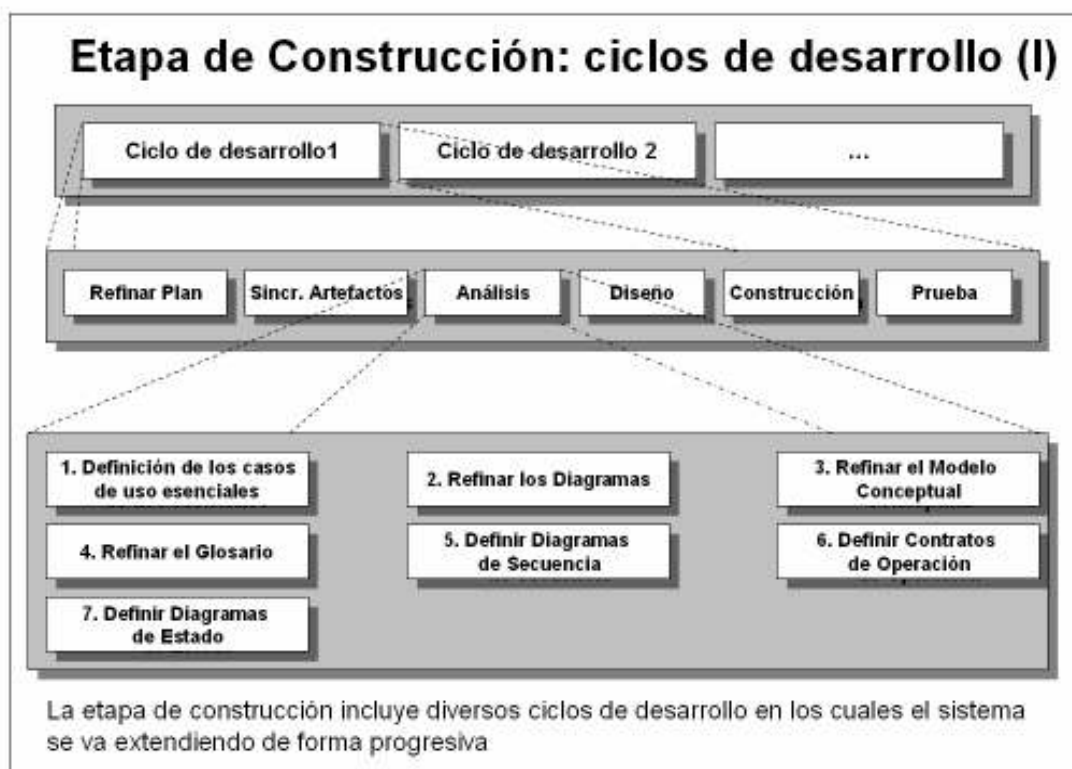


Figura: 3-7 Etapa de Construcción: Ciclos de desarrollo (I)

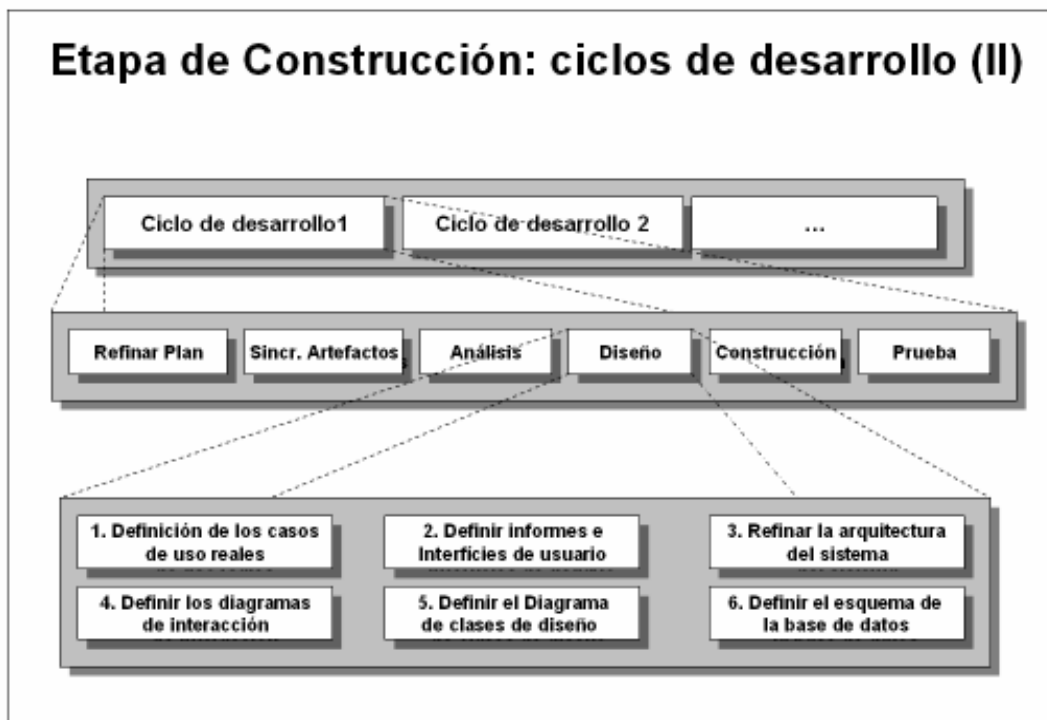


Figura: 3-8 Etapa de Construcción: Ciclos de desarrollo (I)

3.3 Aplicación de la metodología UP en el proyecto

Indicaremos a continuación la correspondencia entre las etapas del Proceso Unificado y los apartados de la memoria.

- **Concepción:** corresponde a los apartados de Introducción y Análisis.
- **Elaboración:** corresponde a los apartados de Análisis.
- **Construcción:** dentro de esta etapa por ser iterativo, se lleva a cabo las cuatro etapas más importantes de un proyecto que corresponde con los apartados Análisis, Diseño, Implementación y Pruebas.
- **Transición:** corresponde al apartado de Implantación y mantenimiento.

A lo largo de la memoria no todos los diagramas se utilizarán en el proyecto, debido a que Drupal es un CMS no orientado a objeto y no sirve mucho para su propósito. Se intentará utilizar el mayor número posible.

Todo el proceso ha sido iterativo e incremental. A partir de cada Caso de Uso se comienza un nuevo ciclo de desarrollo en el cual se itera sobre el análisis, el diseño, la implementación y las pruebas de manera que en cada iteración se refina el trabajo ya realizado con lo que se obtiene un incremento del producto desarrollado que hace que el sistema mejore, alcanzando así los objetivos planteados inicialmente.

El Proceso Unificado a parte de definir que pasos hay que seguir para el desarrollo de Software, también especifica quién debe de realizar estos pasos. En este proyecto todos los rolos han sido desarrollados por mí, menos los del analista que ha sido compartido con Daniel.

4. Análisis

El análisis de requerimientos es la primera etapa de un proyecto software, en ella se trata de definir las condiciones o capacidades necesarias para uno o varios usuarios con el fin de solucionar un problema o conseguir un objetivo.

Para la creación global del sistema se necesita comprender todos los objetivos y necesidades del usuario. En primer lugar, hemos de especificar el comportamiento externo del sistema desde el punto de vista del usuario, en forma de requisitos. La determinación de los requerimientos se haya en base a la experiencia, de hablar con los usuarios finales sobre sus necesidades y/o analizando un sistema de software existente.

Hay dos tipos de requerimientos: **funcionales** (Que debe hacer el sistema), y **no funcionales** (otros requisitos sobre el entorno (sistema operativo, sistema gestor de base de datos, sistema de archivos, ...), ergonómicos (interfaz gráfica, etc...), de rendimiento, de tiempo, formato de entrega, etc...)

4.1 Definición de actores

El estudio del sistema permitió detectar la existencia de los siguientes actores:

- **Usuario anónimo:** representa a los usuarios que no están dados de alta en nuestro sistema o que aún no han “logueado” (entrar en el sistema con usuario y un password que previamente se registro).
- **Usuario registrado:** representa a los usuarios que han entrado en el sistema.
- **Administrador:** representa al administrador del sistema, tiene control total sobre el sistema. En Drupal el primer usuario que se registra en el sistema es el administrador.

4.2 Requerimientos funcionales

A continuación definiremos los requerimientos necesarios para cada tipo de actores.

Usuario anónimo:

- Poder consultar en la base de datos las encuestas y los usuarios ya registrados.
- Poder votar en las encuestas.
- Poder registrarse en el sistema.
- Poder consultar a las encuestas más votadas.
- Poder acceder al folksonomía¹.
- Poder consultar a los líderes de opinión, estos son los usuarios registrados que tienen mayor número obtenido en restar los votos a favor y los votos en contra que se le otorgan los usuarios en los comentarios.
- Poder votar comentario.
- Poder votar la calidad de la encuesta.
- Poder consultar a las encuestas relacionadas a la encuesta que se esta viendo.
- Poder compartir la encuesta con otra persona (enviar la encuesta a un amigo).

Usuario registrado:

- Todos los requerimientos del usuario anónimo.
- Poder dar su opinión sobre las encuestas (posibilidad de comentar).
- Poder crear nuevas encuestas.
- Modificar su perfil (incluye sus datos personales, subir su propio avatar).

Administrador:

- Determinar las encuestas más votadas sean del día, la semana o mes.
- Determinar la frecuencia de recalcu para las estadísticas de las encuestas más votadas, los líderes de opiniones.
- Determinar las categorías de la web.
- Determinar el orden en los comentarios, si es por fecha de creación o por los puntos que tienen (puntos positivos menos los negativos).
- Cerrar encuestas.
- Eliminar encuestas, comentarios.

¹ Calco del inglés folksonomy, es un neologismo que da nombre a la categorización colaborativa por medio de etiquetas simples en un espacio de nombres llano, sin jerarquías ni relaciones de parentesco predeterminadas. Ejemplo en el apartado Glosario

4.2.1 Diagramas de casos de uso

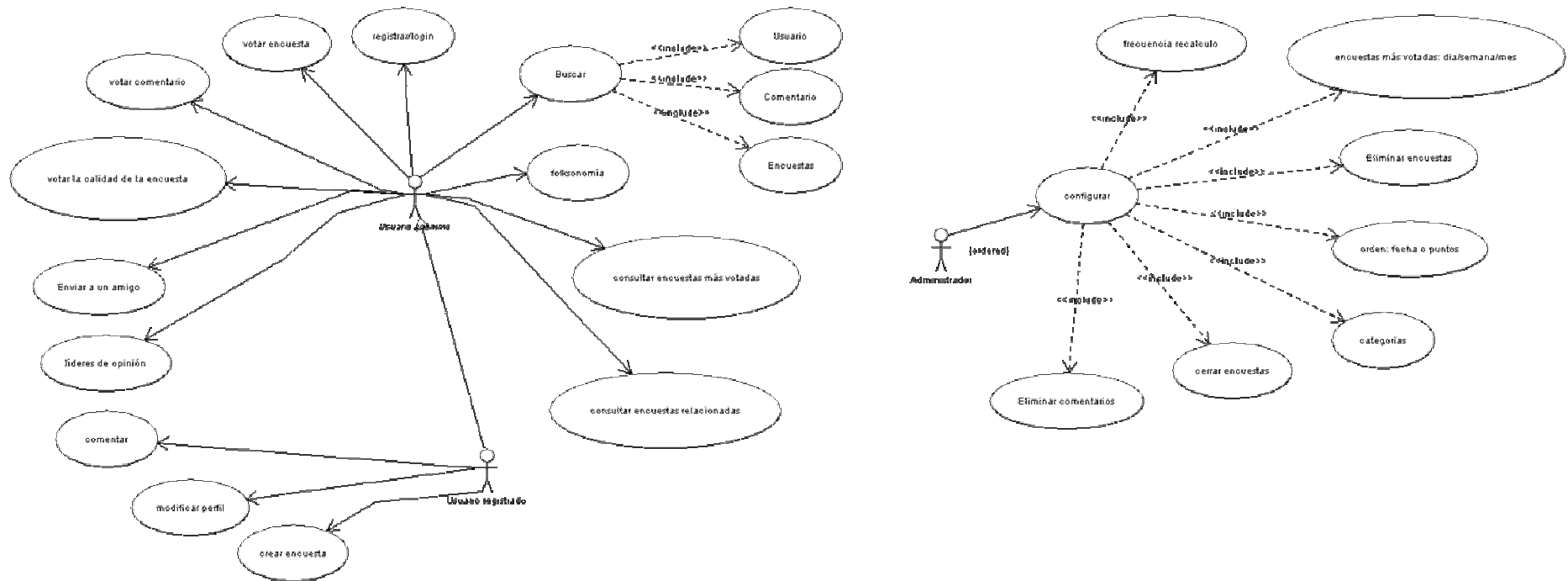


Figura: 4-1 Diagrama de casos de uso

4.2.2 Especificación de los casos de Uso

Caso de Uso:	Buscar	
Versión	1.0	
Autores	SP	
Descripción	Los usuarios podrán buscar en la base de datos algunos datos importantes como encuestas, usuarios o comentarios.	
Actores	Usuario anónimo, Usuario registrado	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando uno de los usuarios quiere buscar algún dato en el sistema.	
		2.- El sistema genera un campo de texto para que el usuario pueda introducir el texto a buscar y un botón.
	3.- El usuario introduce el texto y aprieta el botón.	
		4.- Se busca en la base datos.
		5.- El sistema muestra una página con el resultado de la búsqueda.
Requerimientos no funcionales	El campo de texto y el botón tiene que ir en la parte derecha de la cabecera del web.	
Prioridad	Normal.	

Caso de Uso:	Registrar/Login	
Versión	1.0	
Autores	SP	
Descripción	El usuario desea registrarse en el sistema o logear en el sistema.	
Actores	Usuario anónimo.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario anónimo quiere registrarse.	
		2.- El sistema genera una página con el formulario y

		los campos necesarios para añadir al usuario dentro del sistema.
	3.- El usuario introduce los datos que se le piden.	
		4.- Se comprueban los datos rellenos por el usuario.
		5.- Se registran en la base de datos.
		6.- Se envía un email con los datos al usuario.
	7.- El usuario recibe por la pantalla la confirmación del alta.	
	Flujos alternativos	Acciones de los actores
1.- El usuario quiere logear.		
		2.- El sistema genera una página donde el usuario debe de introducir su identificador de usuario y su password.
3.- El usuario introduce los datos que se le piden.		
		4.- Se comprueban los datos rellenos por el usuario.
5.- El usuario recibe por la pantalla la confirmación de que ya esta logeado.		
Acciones de los actores		Respuesta del sistema
		5.- Algunos de los datos introducidos son incorrectos.
6.- El usuario recibe por la pantalla con el texto del error y el campo afectado.		
Prioridad		Normal.

Caso de Uso:	Votar encuesta	
Versión	1.0	
Autores	SP	
Descripción	El sistema tiene que permitir a cualquier tipo de usuario poder votar en las encuestas.	
Actores	Usuario anónimo, usuario registrado.	
Precondición	La encuesta que se desea votar debe de existir y que este activa.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario quiere votar una encuesta.	
		2.- Se comprueba que el usuario no ha votado anteriormente.
		3.- El sistema genera una encuesta.
	4.- El usuario elige la opción de su preferencia.	
		4.- Se registra la opción elegido juntos con sus datos en la base de datos.
		5.- Recalcula los totales de la encuesta y se genera la encuesta con el nuevo resultado.
6.- El usuario recibe por la pantalla la confirmación del voto.		
Flujos alternativos	Acciones de los actores	Respuesta del sistema
		3.- Usuario ya votado.
	4.- El usuario recibe por la pantalla el mensaje de que anteriormente ya había votado.	
Prioridad	Normal.	

Caso de Uso:	Votar comentario	
Versión	1.0	
Autores	SP	
Descripción	El usuario después de leer un comentario de otro usuario registrado desea expresar su desacuerdo o su aprecio	

	hacia el comentario.	
Actores	Usuario anónimo, usuario registrado.	
Precondición	La encuesta y el comentario deben de existir para poder votar. Además la encuesta debe de estar activa.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario quiere votar el comentario.	
		2.- El sistema genera la encuesta juntos con los comentarios que han publicado los usuarios.
		3.- Para evitar spam, se deja un intervalo de tiempo para poder volver a votar. Por eso el sistema verifica si el usuario ya ha pasado ese intervalo.
	3.- El usuario leer el comentario y expresa su opinión dando un voto positivo o negativo, si esta conforme con el comentario o no.	
		4.- El sistema registra su voto.
		5.- Se recalcula los totales de votos a favor y en contra.
	6.- El usuario recibe por la pantalla los datos de la votación.	
Flujos alternativos	Acciones de los actores	Respuesta del sistema
		4.- No permitimos al usuario votar. Porque aún no ha pasado el tiempo requerido.
Prioridad	Normal.	

Caso de Uso:	Votar la calidad de la encuesta
Versión	1.0
Autores	SP

Descripción	Para obtener la opinión del usuario sobre la calidad de la encuesta que se formuló otro usuario.	
Actores	Usuario anónimo, Usuario registrado.	
Precondición	La encuesta debe de existir y estar activa.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario quiere votar sobre la calidad de la pregunta de la encuesta.	
		2.- El sistema genera la encuesta junto con un sistema de estrella para poder votar. Cuantos más estrellas mejor puntuación.
	3.- El usuario vota.	
		4.- Se registra el voto en la base de datos. Si ya existía un voto, se sobrescribe con el valor de este último.
		5.- Se calcula los votos totales que tiene.
	7.- El usuario ve la actualización de votos.	
Prioridad	Normal.	

Caso de Uso:	Enviar a un amigo	
Versión	1.0	
Autores	SP	
Descripción	El usuario desea compartir la encuesta con un amigo suyo, ya que ve que la encuesta es muy interesante.	
Actores	Usuario anónimo, usuario registrado.	
Precondición		
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario quiere compartir la encuesta con un amigo.	
		2.- El sistema genera una página con un formulario para que el usuario lo rellene.
	3.- El usuario introduce las	

	direcciones de correos de sus amigos y el suyo propio junto con un texto que es optativo.	
		4.- Se comprueban los datos rellenos por el usuario sean correcto.
		5.- Se envía un email junto con la url de la encuesta y el comentario opcional del usuario.
	6.- El usuario recibe por la pantalla la confirmación del envío.	
Flujos alternativos	Acciones de los actores	Respuesta del sistema
		5.- Algunos datos son incorrectos.
	6.- El usuario recibe por la pantalla el error.	
Prioridad	Normal.	

Caso de Uso:	Líderes de opinión	
Versión	1.0	
Autores	SP	
Descripción	El sistema muestra los líderes de opinión según el número de votos totales que se obtuvo restando los votos positivos con los votos negativos. Pueden estar clasificados por categoría o totales.	
Actores	Usuario anónimo, usuario registrado.	
Precondición		
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario está en la página principal.	
		2.- El sistema muestra los 5 líderes de opinión con mayor cantidad de votos totales de todas las categorías.
Flujos alternativos	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el	

	usuario está en una categoría definida por el administrador.	
		2.- El sistema muestra los 5 líderes de opinión con mayor cantidad de votos totales según la categoría que se encuentre.
	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario está en una encuesta.	
	2.- El sistema muestra los 5 líderes de opinión con mayor cantidad de votos totales sumando todos los votos en todas las categorías que hayan comentado en la encuesta.	
Requerimientos no funcionales	Se mostrará en las categorías y en las encuestas en el lado izquierdo de la web e inmediatamente después de la cabecera. En el caso que se encuentre en la página principal se mostrará debajo de la encuesta más votada.	
Prioridad	Normal.	

Caso de Uso:	Comentar	
Versión	1.0	
Autores	SP	
Descripción	El usuario desea expresar su opinión después de haber votado en la encuesta.	
Actores	Usuario registrado.	
Precondición	La encuesta debe de existir para poder comentar.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario registrado quiere opinar sobre la encuesta.	
		2.- El sistema genera un formulario al final de los comentarios.
	3.- Se rellena el formulario	

	con la opinión del usuario.	
		4.- Se comprueban los datos rellenos por el usuario.
		5.- Se registran en la base de datos el comentario.
		6.- Se monta el sistema para que pueda votar en el comentario.
	7.- El usuario recibe por la pantalla el comentario recién creado junto con el resto de comentarios ordenado según la fecha de creación o la cantidad de votos.	
Flujos alternativos	Acciones de los actores	Respuesta del sistema
		5.- Se ha introducido algún dato erróneo en el formulario.
	6.- El usuario recibe por la pantalla el error junto con el campo erróneo.	
Prioridad	Normal.	

Caso de Uso:	Modificar perfil.	
Versión	1.0	
Autores	SP	
Descripción	El usuario desea modificar su propio perfil (password, avatar, ...)	
Actores	Usuario registrado.	
Precondición		
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario quiere modificar su propio perfil.	
		2.- El sistema genera una página con el formulario y con los datos actuales de su perfil.
	3.- El usuario introduce los datos que se quiera	

	cambiar.	
		4.- Se comprueban los datos rellenos por el usuario.
		5.- Se cambia los datos en la base datos.
	7.- El usuario recibe por la pantalla la confirmación del cambio.	
Flujos alternativos	Acciones de los actores	Respuesta del sistema
		5.- Algún dato relleno es incorrecto.
	6.- El usuario recibe por la pantalla el error y el campo erróneo.	
Prioridad	Normal.	

Caso de Uso:	Crear encuesta	
Versión	1.0	
Autores	SP	
Descripción	El usuario registrado desea crear su propia encuesta para que el resto de usuario pueda votar.	
Actores	Usuario registrado.	
Precondición		
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario quiere crear una nueva encuesta.	
		2.- El sistema genera una página con el formulario y los campos necesarios para añadir una nueva encuesta.
	3.- El usuario introduce los datos que se le piden.	
		4.- Se comprueban los datos rellenos por el usuario.
		5.- Se registran en la base de datos.
	6.- El usuario recibe por la pantalla la nueva encuesta	

	preparada para votar.	
Flujos alternativos	Acciones de los actores	Respuesta del sistema
		5.- Algún dato rellenado es incorrecto.
	6.- El usuario recibe por la pantalla el error y el campo erróneo.	
Prioridad	Normal.	

Caso de Uso:	Consultar encuesta relacionada	
Versión	1.0	
Autores	SP	
Descripción	El sistema muestra las encuestas relacionadas cuando un usuario esta viendo una encuesta en concreto.	
Actores	Usuario anónimo, usuario registrado	
Precondición	La encuesta debe de existir.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario esta en la página de alguna encuesta.	
		2.- El sistema muestra las encuestas relacionadas según la semblanza de la temática.
Requerimientos no funcionales	Siempre se mostrará en el lado derecha e inmediatamente después de la cabecera.	
Prioridad	Normal.	

Caso de Uso:	Consultar encuesta más votada.	
Versión	1.0	
Autores	SP	
Descripción	El sistema muestra en la página principal y en las categoría las encuestas más votadas.	
Actores	Usuario anónimo, usuario registrado.	
Precondición	Debe de existir encuestas en el sistema.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario se encuentra en la página principal.	
		2.- El sistema genera la encuesta más votada

		durante el día o semana o mes, según lo que marque el administrador. La frecuencia de refresco también lo configura el administrador.
	3.- El usuario recibe por la pantalla la gráfica de la encuesta más votada.	
Flujos alternativos	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario se encuentra alguna categoría.	
		2.- El sistema genera la encuesta más votada. Juntos con las encuestas más votadas de la categoría que esté el usuario.
	3.- El usuario recibe por la pantalla la gráfica de la encuesta más votada junto con una lista de las encuestas más votadas según la categoría que se encuentre.	
Prioridad	Normal.	

Caso de Uso:	Folksonomía	
Versión	1.0	
Autores	SP	
Descripción	El sistema muestra una nube de tags que los usuarios introdujeron durante la creación de las encuestas.	
Actores	Usuario anónimo, Usuario registrado.	
Precondición	Debe de existir encuestas en el sistema y los usuarios hayan introducido tags.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el usuario se encuentra en el menú principal, en las categorías o en la	

	encuesta.	
		2.- El sistema obtiene todos los tags y lo muestra los tags más repetidos con un tamaño de fuente mayor.
	3.- El usuario recibe por la pantalla el nube de tags.	
Requerimientos no funcionales	Las nubes de tags siempre se mostrarán en la parte izquierda de la web.	
Prioridad	Normal.	

Caso de Uso:	Configurar.	
Versión	1.0	
Autores	SP	
Descripción	El administrador necesita configurar algunos parámetros para que el sistema pueda funcionar correctamente y mantenimiento de las encuestas y los comentarios (se alguna encuesta o comentario se sale de tono, el administrador tiene privilegio de borrarlo).	
Actores	Administrador.	
Flujo principal	Acciones de los actores	Respuesta del sistema
	1.- El caso de uso comienza cuando el administrador quiere cambiar algún parámetro.	
		2.- El sistema genera una página con una lista de los parámetros configurable. (frecuencia recalcular, mostrar las encuesta más votadas del día, semana o mes, orden de los comentarios, categorías)
	3.- El administrador selecciona uno de los parámetros.	
		4.- El sistema genera una nueva página con un campo de texto y su descripción del parámetro a cambiar.
	5.- El administrador	

	introduce rellena el dato.		
		6.- Se verifica el dato.	
		7.- Se actualiza el valor.	
	8.- El administrador recibe por pantalla la confirmación del cambio.		
Subflujos			
Flujos alternativos	Acciones de los actores	Respuesta del sistema	
	1.- El administrador desea administrar los comentarios		
		2.- El sistema genera una página con una lista de todos los comentarios. Y unos campos para poder eliminar el comentario.	
	3.- El administrador selecciona el comentario a eliminar y rellena los campos.		
		4.- Se comprueban los datos rellenos por el administrador.	
		5.- Se elimina de la base de datos el comentario.	
	5.- El usuario recibe por la pantalla la confirmación de la eliminación.		
	Acciones de los actores		
	Respuesta del sistema		
	1.- El administrador desea administrar las encuestas		
		2.- El sistema genera una página con una lista de todas las encuestas. Y unos campos para poder eliminar o cerrar la encuesta.	
	3.- El administrador selecciona la encuesta a eliminar/cerrar y rellena los campos.		
		4.- Se comprueban los datos rellenos por el	

		administrador.
		5.- Se elimina o se cierra la encuesta de la base de datos.
	5.- El usuario recibe por la pantalla la confirmación.	
	Acciones de los actores	Respuesta del sistema
3.- El administrador no selecciona ninguna opción.		
Prioridad	Normal.	

4.3 Requerimientos no funcionales

Una vez vistos los requerimientos funcionales pasamos a enumerar qué otros aspectos debemos tener en cuenta a la hora de diseñar la aplicación.

- Compatible con cualquier navegador.
- Título de la página web: www.rankitt.com
- Resolución de la página web: 1024 x 768
- Lenguaje de programación PHP, CSS y HTML.
- Uso del servidor web Apache.
- Uso de MySql como sistema gestor de bases de datos SQL.
- Categoría de la web:
 1. Cinema
 2. Music
 3. Books
 4. Internet
 5. Politics
 6. Gossip
 7. Travel
 8. Economy
 9. Sports
 10. Psychology
 11. Other

4.4 Planificación

Para comenzar con el proceso de desarrollo de un sistema Software es necesario definir una planificación inicial. Esta planificación debe ser realista, estructurada y ajustada a los recursos disponibles. En definitiva, nos ayuda a organizar la metodología de trabajo a seguir para alcanzar los objetivos planteados.

Para representar las fases y actividades de la estructura analítica del proyecto, se ha usado los diagramas de Gantt. El objetivo de dicho diagramas es el de mostrar el tiempo de dedicación previsto para diferentes tareas a lo largo de un tiempo total determinado.

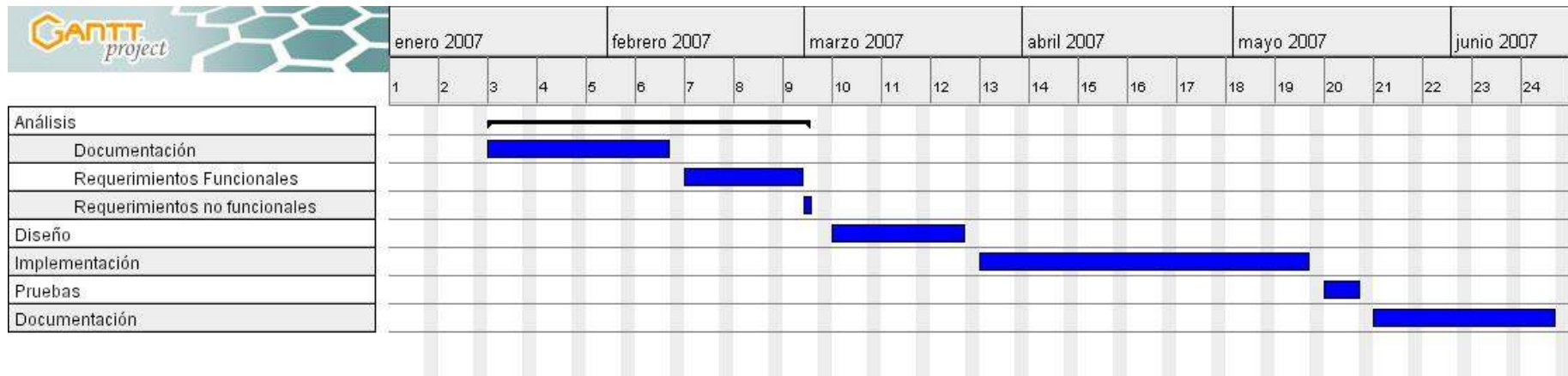


Figura: 4-2 Planificación temporal del proyecto.

5. Diseño e Implementación

En el capítulo anterior hemos estado definiendo los diferentes objetivos que debía de cumplir este proyecto. En esta sección explicaremos el diseño llevado a cabo para crear la web rankitt y su posterior implementación.

En la fase de diseño se establecerá el comportamiento dinámico del sistema, es decir, como debe reaccionar ante los acontecimientos. El resultado obtenido de la etapa de diseño facilita enormemente la implementación posterior de nuestro sistema, pues proporciona la estructura básica del sistema y como los diferentes componentes actúan y se relacionan entre ellos.

5.1 Diseño de datos

Para cualquier web social es muy importante almacenar los datos de la web, para ello es necesario diseñar la estructura de la base de datos que va a contener esos datos.

Drupal al ser un CMS dispone de un diseño predefinido de la estructura de base de datos. Para nuestra necesidad tenemos que crear algunas tablas para poder guardar informaciones importantes para poder trabajar con ellos después.

Según se observa en la figura 5-1, la estructura de base de datos de Drupal se centra en dos tablas que son: **users**, **node**. Que forman el epicentro y tienen más relación con otras tablas. Estos nos indican que los usuarios y los contenidos son la parte más importante en un CMS como Drupal. Según el análisis de requerimientos del apartado anterior, al disponer de tres clases muy diferentes de actores, nos obliga a guardar los datos de los usuarios registrados y el administrador en la base de datos para posteriormente poder recuperar. La tabla users de Drupal nos proporciona los campos necesarios para llevar a cabo todo el tema relacionado con los usuarios. Al estar organizado modularmente si tuviéramos que necesitar más datos de los usuarios solamente haría falta descargar el módulo "*profile*". Este módulo permite al administrador definir nuevos campos que los usuarios deberán de rellenar cuando se dé de alta en el sistema.

La división en categoría, es uno de los requerimientos no funcionales, se salva con la utilización de las tablas: **term_node**, **term_data**. En el term_node estará definida la relación entre la encuesta y la categoría correspondiente. Y en term_data se guardará las 11 categorías que se definió al principio.

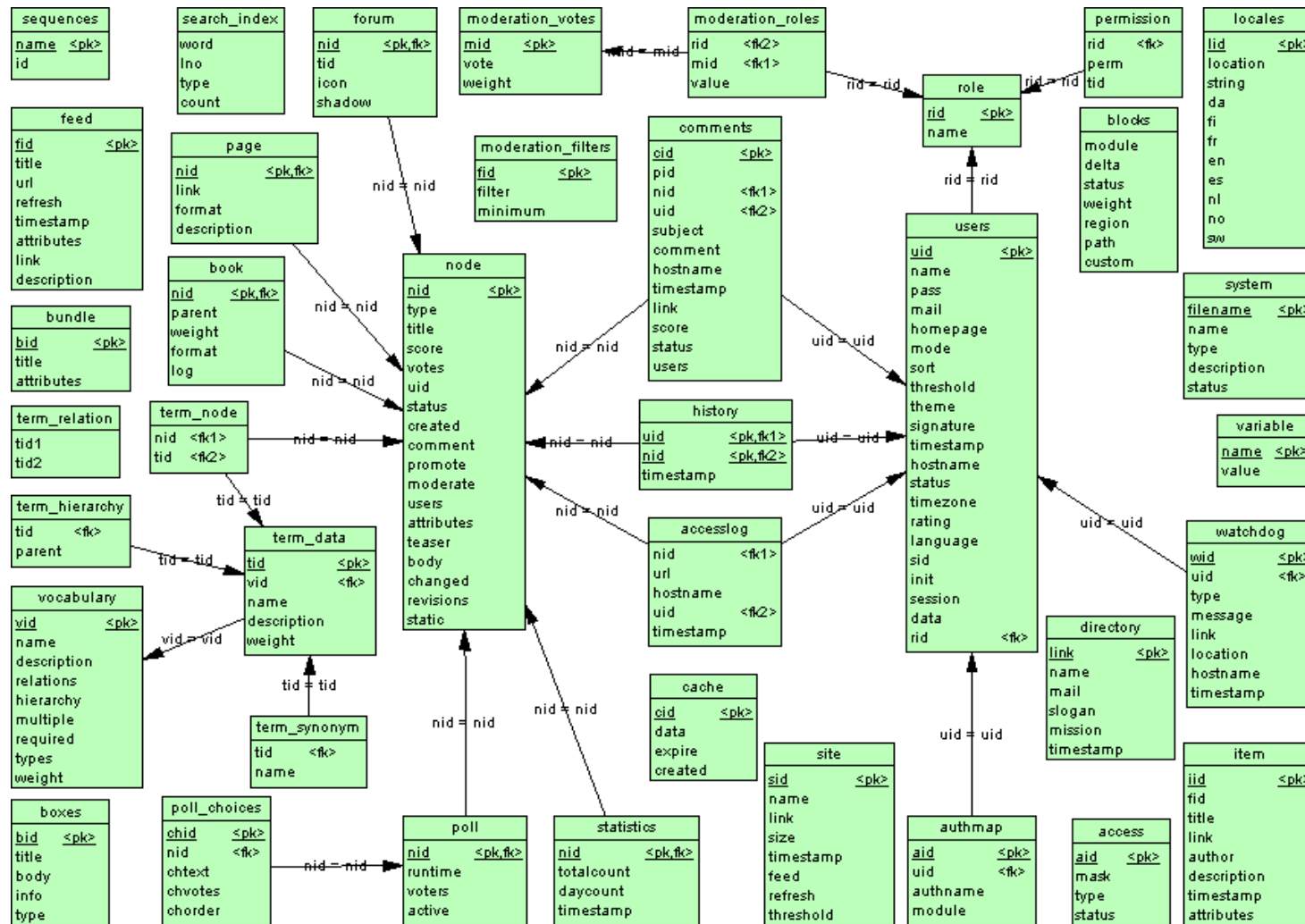


Figura: 5-1 Estructura de base de datos predefinida de Drupal.

Las encuestas y los comentarios se guardan en la tabla **node** y en **comment**. Se guardarán la información más relevante como título, el cuerpo, la fecha de creación, el usuario que lo creo, etc ...

Para la parte importante de la web, los votos se ha necesitado crear tres tablas nuevas más un módulo llamado "*VotingApi*" que nos ha servido como base para votar. Si se fija en la figura 5-2, las nuevas tablas son Advpoll, Advpoll_Choices y calculate_vote. Además de las tablas que se necesita para el correcto funcionamiento de "*VotingApi*".

Normalmente en Drupal las tablas se llaman igual que el módulo, pero en el caso de que sea necesario más de una tabla, se le asignar un prefijo según la acción que se vaya a desarrollar. En nuestro caso el módulo encargado de permitir a los usuarios crear y votar en las encuestas se llama Advpoll y por eso las dos tablas comienzan por Advpoll.

Como comentamos anteriormente en node se guardaban la información de la encuesta. Concretamente una parte de la información que compone una encuesta, en Advpoll se va a guardar el resto de la información para el tratamiento interno. A continuación explicamos los campos más importantes: el nid nos permite enlazar con el nodo correspondiente; el campo mode sirve para diferencia si es una encuesta o un ranking; active indica si la encuesta se encuentra activo.

En la tabla Advpoll_Choices se guarda las opciones de las encuestas.

Para guardar los votos se ha utilizado las tablas y los API (Application Programming Interface) del módulo "*VotingApi*", es la base de todos los módulos que quieren trabajar con los votos. Tiene más tablas pero en la figura solo mostramos las dos más importante y que nos afecta a nuestra aplicación. En votingApi_vote se guardan los votos individuales que realizan los usuarios del sistema. Junto con el voto se guarda también timestamp, el host desde donde se ha conectado el usuario. VotingApi_cache como su nombre indica es un cache, nos devuelve los votos totales de la encuesta y también los votos totales positivos o negativos para los comentarios.

La tabla Calculate_vote, es muy parecido a VotingApi_cache en cuando a funcionalidad. Es un cache que se aloja las 10 encuestas de cada categoría con más votos durante el día. También se calcula las semanales y los mensuales. Los líderes de opinión también son guardados en esta tabla.

Estás se relacionan con la tabla node o coment dependiendo del valor en el campo content_type y el valor content_id que es el identificador único para los nodos y comentarios.

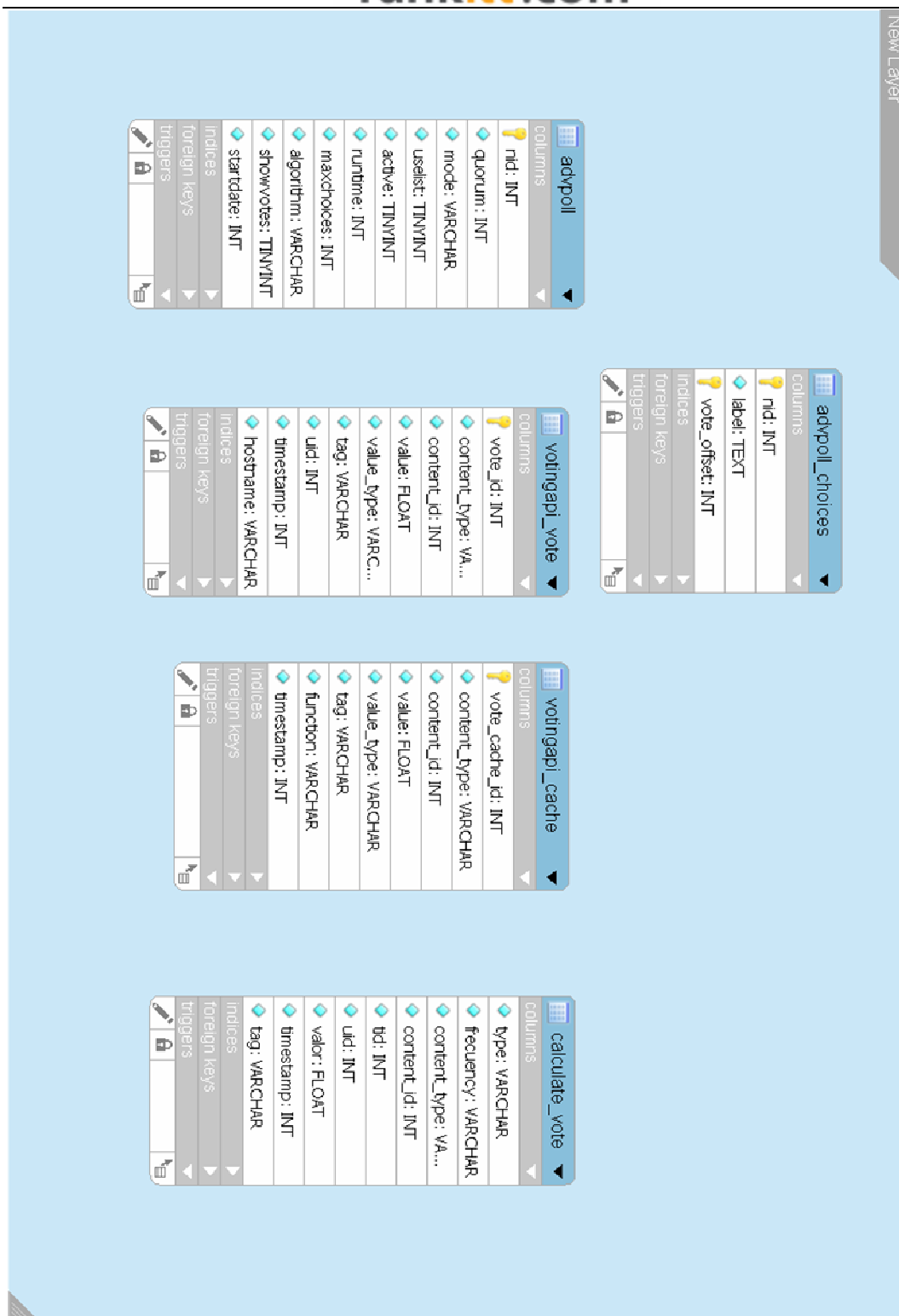


Figura: 5-2 Diseño de la tabla que se ha tenido que crear. .

5.2 Creación de tablas en Drupal

En Drupal crea una tabla automáticamente si anteriormente hemos definido en el fichero "modulo.install", modulo es el módulo a instalar, las sentencias SQL para la creación de la tabla. Este fichero debe de situarse en el mismo directorio donde esté definido las funciones del módulo que tiene una

nomenclatura siguiente “*modulo.module*”, su extensión por defecto es *module*, aunque internamente es código PHP.

Dentro del fichero “*modulo.install*” tiene que estar definido la función *modulo_install()*. Drupal cuando se activa un nuevo módulo se llama a esta función para crear las tablas. Drupal al ser un sistema que soporta varios gestores de base de datos permite instrucciones para ellos. En la variable `$GLOBALS['db_type']` se obtiene el tipo.

Drupal proporciona unos API's para acceder a la base de datos.

- ***db_query()***: ejecuta una query básica en la base de datos. Tiene un parámetro string que contiene un SQL query.
- ***pager_query()***: ejecuta una query más compleja, devuelve una query paginable, cada página tiene un determinado número de componentes que es indicado a través del parámetro.
- ***tablesort_sql()***: sirve para generar una query de una tabla ordenable por sus campos.
- ***db_fetch_object()***: devuelve una fila de la consulta previa que se ejecutó con *db_query()* y sus equivalentes.
- ***db_num_rows()***: devuelve el número de filas que se encontró.

Hemos explicado las funciones típicas que utilizan en Drupal para el acceso a la base de datos. Para profundizar recomiendo leer los API de drupal².

Ejemplos: *calculatevote.install*

```
<?php
// $Id: calculatevote.install

/**
 * Implementation of hook_install().
 */
function calculatevote_install() {
  switch ($GLOBALS['db_type']) {
    case 'mysql':
    case 'mysqli':
      db_query("CREATE TABLE {calculate_vote} (
        `type` VARCHAR(20) NOT NULL,
        `frequency` VARCHAR(20) NOT NULL,
        `content_type` varchar(20) NOT NULL,
        `content_id` int(10) NOT NULL,
        `tid` int(10) default 0,
        `uid` int(10) NOT NULL,
        `valor` float(10) default NULL,
        `timestamp` int(11) default NULL,
        `tag` varchar(20) default 'poll'
      ) ENGINE=MyISAM /*!40100 DEFAULT CHARACTER SET utf8 */");

      break;
  }
}
/**
```

² Drupal API: <http://api.drupal.org/api/group/database/4.7>

```
* Implementation of hook_uninstall().
*/
function calculatevote_uninstall() {
    db_query('DROP TABLE {calculate_vote}');
    variable_del('calculatevote_interval');
}
```

Ejemplo: advpoll.install

```
<?php
// $Id: advpoll.install

/**
 * Implementation of hook_install().
 */
function advpoll_install() {
    switch ($GLOBALS['db_type']) {
        case 'mysql':
        case 'mysqli':
            db_query("CREATE TABLE {advpoll} (
                `nid` int(10) NOT NULL,
                `quorum` int(10) NOT NULL,
                `mode` varchar(32) NOT NULL,
                `uselist` tinyint default '0',
                `active` tinyint default '1',
                `runtime` int NOT NULL default '0',
                `maxchoices` int unsigned NOT NULL default '0',
                `algorithm` VARCHAR(100),
                `showvotes` tinyint,
                `startdate` int unsigned,
                PRIMARY KEY (`nid`)
            ) ENGINE=MyISAM /*!40100 DEFAULT CHARACTER SET utf8 */");

            db_query("CREATE TABLE {advpoll_choices} (
                `nid` int(10) NOT NULL,
                `label` text NOT NULL,
                `vote_offset` int(2) unsigned default NULL,
                PRIMARY KEY (`nid`, `vote_offset`),
                KEY `vote_offset` (`vote_offset`)
            ) ENGINE=MyISAM /*!40100 DEFAULT CHARACTER SET utf8 */");

            break;
        }
    }

/**
 * Implementation of hook_uninstall().
 */
function advpoll_uninstall() {
    db_query("DELETE FROM {votingapi_vote} WHERE
        content_type='advpoll'");
    db_query('DROP TABLE {advpoll}');
    db_query('DROP TABLE {advpoll_choices}');
    variable_del('advpoll_default_mode');
}
```

5.3 Diseño de la Interfaz

Al ser una parte importante de una web social, tuvimos varias reuniones con Daniel Martínez para llegar a un acuerdo sobre las necesidades de rankitt para

atraer al usuario, para ello había que tener en cuenta la usabilidad en la web.

La usabilidad de una web viene determinada por cinco atributos definidos por Shneiderman:

- **Facilidad de aprendizaje:** Se entiende el tiempo necesario para que el usuario novato pueda asimilar las tareas básicas de un sistema con el cual no había tenido contacto previamente.
- **Velocidad de desempeño:** Una vez que el usuario ha aprendido a utilizar el sistema, se va a ponderar el lograr la velocidad con que puede completar una tarea específica.
- **Tasa de error por parte de los usuarios:** Este apartado apunta hacia los errores que comete el usuario al utilizar el sistema. Una aplicación ideal evitaría que el usuario cometiera errores y funcionaria de manera óptima a cualquier petición por parte del usuario. En la práctica esto difícilmente se logra. Es vital que una vez que se produzca un error el sistema se lo haga saber rápida y claramente al usuario, le advierta de la severidad del mismo y le provea de algún mecanismo para recuperarse de ese error.
- **Retención sobre el tiempo:** Cuando un usuario ha utilizado un sistema tiempo atrás, y tiene la necesidad de utilizarlo de nuevo la curva de aprendizaje debe de ser significativamente menor que el caso del usuario que nunca haya utilizado dicho sistema. Esto es de primordial importancia para aplicaciones usadas intermitentemente.
- **Satisfacción subjetiva:** Este atributo se refiere a la impresión subjetiva del usuario respecto al sistema.

Por ello definimos las estructuras de las pantallas ajustando al máximo a esas cinco propiedades. Las principales pantallas siguen la misma estructura, con una cabecera donde irá el logo de la web en la parte izquierda y en la parte derecha irá los links para ir al “home”, un link con un mensaje dando la bienvenida al usuario, si éste estuviera registrado y ha entrado en el sistema, sino un link para que el usuario pueda entrar en el sistema, y por último, un link para salir del sistema o un link donde llevará al usuario a una página para registrarse. Debajo de estas estará situado el formulario para buscar una encuesta, un usuario o un comentario.



Figura: 5-3 Cabecera de la web

Después vendrá el contenido que defiere según donde se encuentre el usuario navegando. En el pie de la página irá el copyright junto con unos enlaces de información sobre la web.

La web predomina el color naranja, visualmente atrae más a los usuarios, con letras negras. Los títulos deben de ir en una especie de fondo naranja con las letras blancas.

Para mostrar la gráfica web se ha optado por utilizar la librería grafica GD para crear la gráfica en el cuál finalmente es incrustada en una película flash. En ella aparece una bombilla, que se ilumina cuando el usuario pasa su ratón por encima, por cada opción tenga la encuesta. Si el usuario ya ha votado, no se muestra ninguna bombilla.

Destino preferido para ir de vacaciones

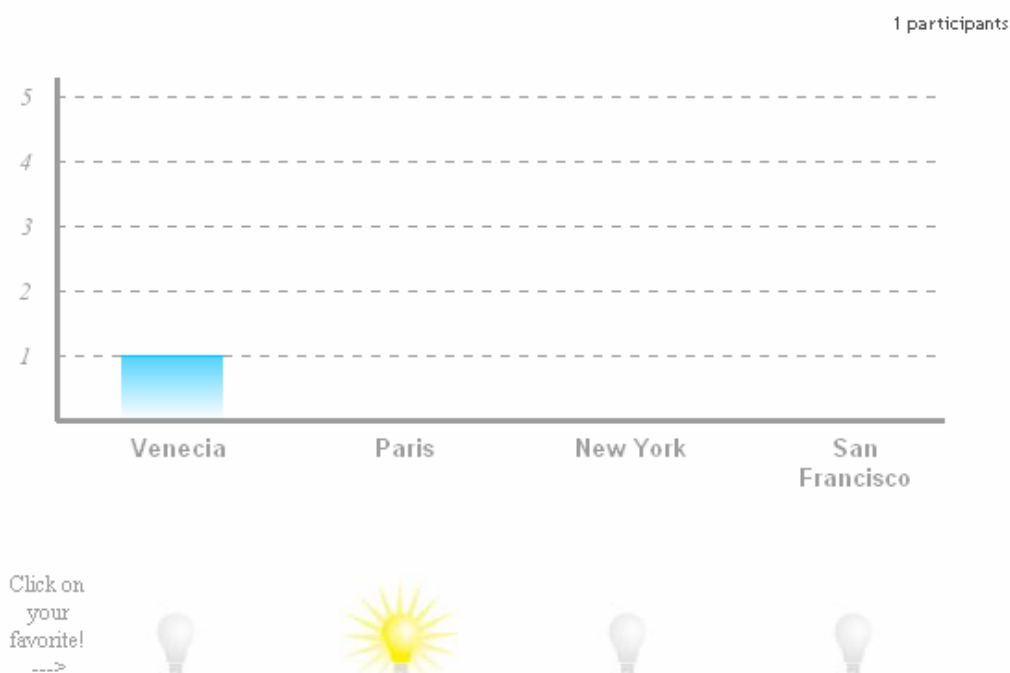


Figura: 5-4 Ejemplo de encuesta

Para poder votar en los comentarios se adapto dos bombillas, una de color rojo, otra de color verde. Representa desacuerdo en el comentario o acuerdo.



Figura: 5-5 Bombillas para votar en los comentarios

Para fomentar la participación, se les muestra a los usuarios encima de la gráfica del voto más votado un texto descriptivo junto con un link para crear una nueva encuesta.

5.4 Diagrama de pantallas

El diagrama de pantallas nos permite esquematizar al detalle la interfaz gráfica y los contenidos de nuestra web.

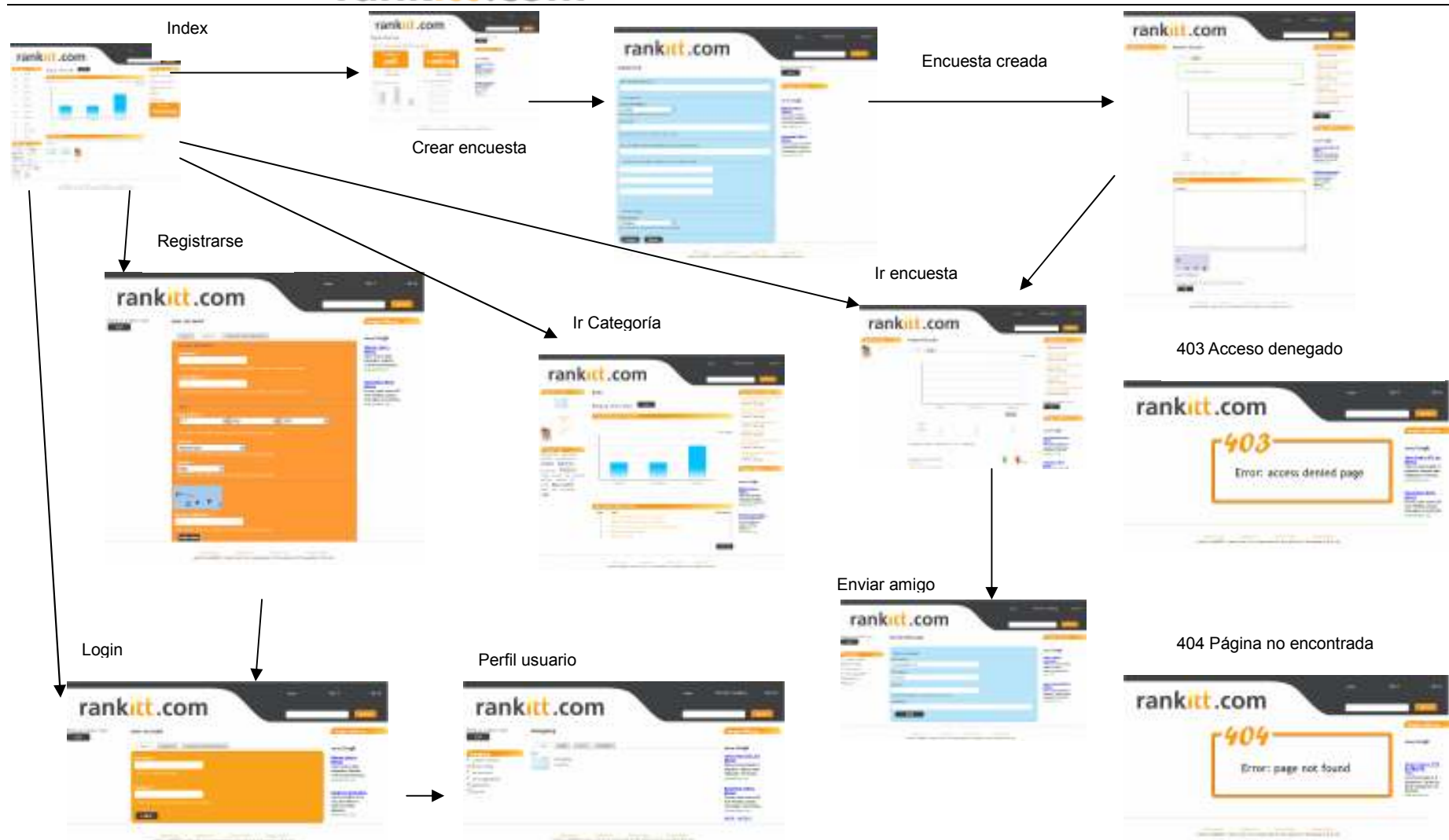


Figura: 5-6 Diagramas de pantallas

5.5 Esquema de la arquitectura

Al ser un sistema basado en la web e Internet el cual tiene que soportar el acceso simultáneo y concurrente de varios usuarios, la arquitectura propuesta es la Arquitectura Cliente-Servidor:

La arquitectura Cliente/Servidor es una nueva tendencia en el desarrollo de redes locales, debido a la necesidad de compartir datos, para poder optimizar los recursos de hardware y software. El paradigma cliente-servidor tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de funciones: el servidor administra la parte que debe ser compartida por varios usuarios, y el cliente maneja sólo lo particular de cada usuario.

Las principales características de la arquitectura son: el cliente presenta a todos sus clientes una interfaz única y bien definida; el cliente no necesita la lógica del servidor, sólo su interfaz externa; el cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo; los cambios en el servidor implican pocos o ningún cambio en el cliente.

Los clientes realizan generalmente funciones como manejo de la interfaz de usuario, captura y validación de los datos de entrada y generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones como gestión de periféricos compartidos, control de acceso concurrente a bases de datos compartidas y enlaces de comunicaciones con otras redes de área local o externa.

Esta arquitectura se puede clasificar en cinco niveles, según las funciones que asumen el cliente y el servidor, tal y como se puede ver en el siguiente diagrama:

Presentación distribuida: El cliente asume parte de las funciones de presentación de la aplicación, ya que siguen existiendo programas en el servidor dedicados a esta tarea. Esta técnica dificulta su mantenimiento en las aplicaciones orientadas a terminales. Además, el servidor ejecuta todos los procesos y almacena la totalidad de los datos.

Presentación remota: La aplicación está soportada directamente por el servidor, excepto la presentación que es totalmente remota y reside en el cliente. Los terminales del cliente soportan la captura de datos, incluyendo una validación parcial de los mismos y una presentación de las consultas.

Proceso distribuido: La lógica de los procesos se divide entre el cliente y el servidor. El diseñador de la aplicación debe definir los servicios y las interfaces del sistema de información de forma que los papeles de cliente y servidor sean intercambiables, excepto en el control de los datos que es responsabilidad exclusiva del servidor

Gestión de datos remota: El cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada.

Bases de datos distribuidos: El reparto de tareas es como en el caso anterior y además el gestor de base de datos divide sus componentes entre el cliente y el servidor. Las interfaces entre ambos están dentro de las funciones del gestor de datos y, por lo tanto, no tienen impacto en el desarrollo de las aplicaciones.

Para el diseño de esta aplicación, se ha elegido la configuración de la arquitectura cliente-servidor con bases de datos distribuidas. El cliente será un navegador Web y se encarga de presentación de los datos y de realizar peticiones al servidor. El servidor recibe, procesa accediendo a los datos en caso necesario y responde a las peticiones con la información demandada por el cliente para que éste la presente correctamente.

5.6 Diagramas de secuencias

Los diagramas de secuencia del sistema toman como punto de partida a los casos de uso y muestran, para cada escenario particular de un caso de uso, los acontecimientos generados por los actores externos, su orden y los acontecimientos internos en el sistema (operaciones) que resultan de su invocación.

Se define un diagrama de secuencia para cada curso relevante de acontecimientos de un caso de uso. Para construir un diagrama de secuencia se siguen los pasos descritos a continuación: Primero dibujar una línea vertical que representa el sistema, segundo dibujar una línea para cada actor que interacciona directamente con el sistema y finalmente del curso de acontecimientos del caso de uso, identificar los acontecimientos externos generados por los actores y mostrarlos en el diagrama.

A continuación se muestra los diagramas de secuencia que hemos tenido que implementar, ya que Drupal no tenía esa funcionalidad que estábamos buscando.

Para simplificar el esquema cuando nos referimos al USUARIO engloba a los usuarios anónimos y a los usuarios registrados. Estos dos usuarios comparten la misma funcionalidad.

Votar encuesta

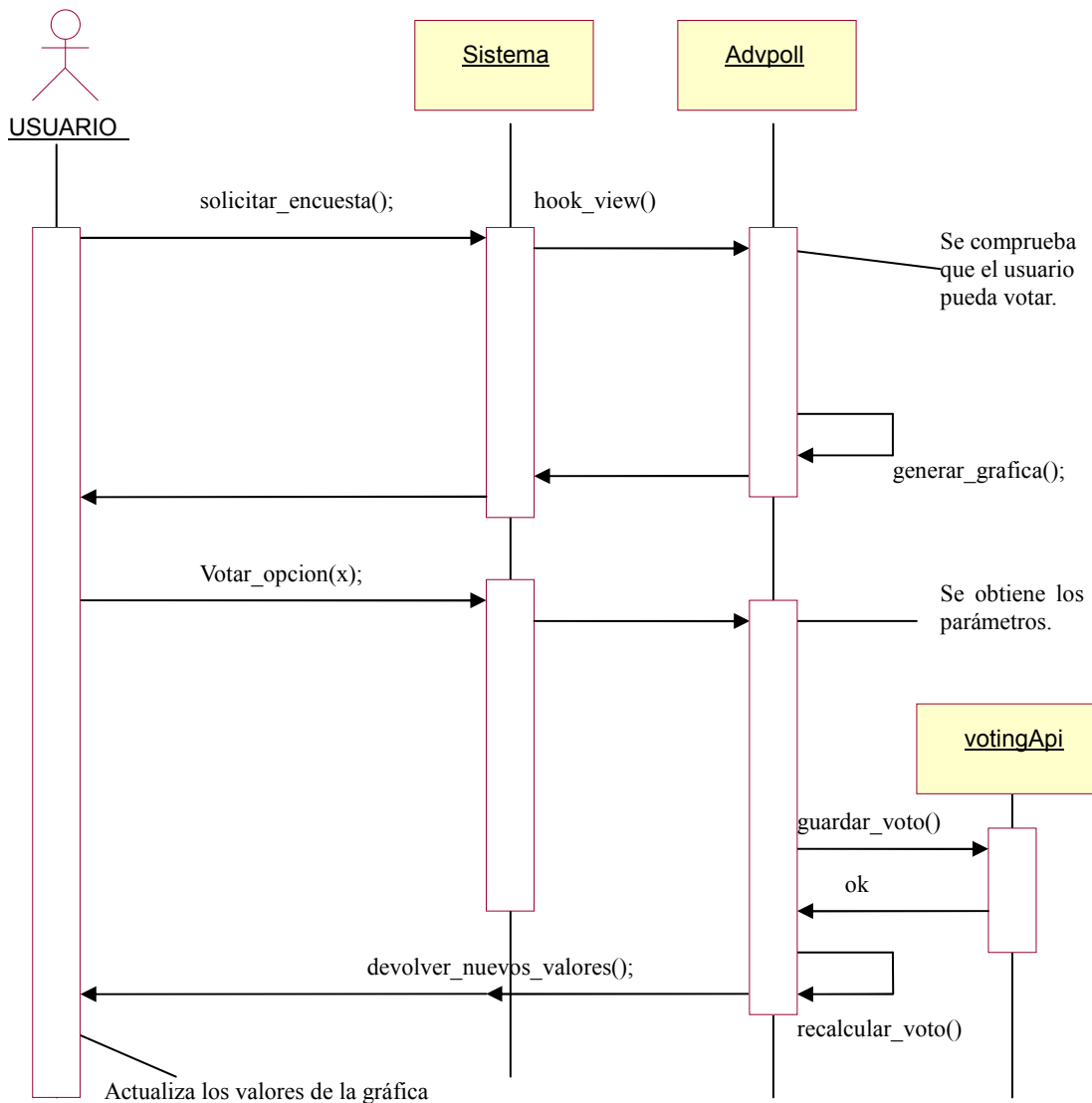


Figura: 5-7 Diagramas de secuencia. Votar encuesta

Implementación en Drupal

Para implementar esta funcionalidad se ha tenido que crear a partir de un módulo base llamado Advpoll. Cuando el usuario llama a **solicitar_encuesta()** en verdad está solicitando en el navegador con una url donde se le pasa el número de encuesta que quiere votar. La forma de la url sería de esta manera, formado al principio de todo el nombre del host, después vendría el directorio donde se ha instalado en el servidor, si está en el directorio base, este valor no tendría en cuenta, después vendría ?q=node/ y el número de nodo.

```
http://host./drupal/?q=node/nid
```

Drupal llama a todos los módulos que tengan implementado el hook_view. En nuestro módulo habíamos implementado la función advpoll_view, y es llamado

por el sistema. Verificamos que nos estamos ante un envío de un voto, eso se consigue accediendo al valor que tiene \$_POST['op']. Para este caso el valor que contiene la variable no es t('Vote'), por lo tanto sigue la ejecución.

El siguiente paso consiste en verificar si el usuario tiene permiso para votar, se comprueba con la siguiente condición:

```
if (!$node->voted && arg(2) != 'results' && $node->active && advpoll_eligible($node))
```

En la variable \$node existe un campo voted que nos indica si el usuario actual ya ha votado, la segunda condición nos indica que no estamos viendo un resultado de la votación. La tercera indica si la encuesta está activa o no y finalmente advpoll_eligible, es una función donde indicará si el usuario tiene permisos del tipo 'vote on polls'. Este es el sistema de permisos de Drupal, solamente los administradores pueden asignar permisos a los distintos tipos de usuarios.

Por último se llama a advpoll_view_voting que nos genera el código HTML en una variable para ser mostrado en el navegador del cliente. Para solventar el problema de colocación de las bombillas decidimos utilizar una película flash, incrustando la imagen que generaba nuestra función php encargada de ello utilizando la librería GD. La película flash está programada en ActionScript, donde a parte de pasarnos la url de la función php de crear la gráfica hay que pasarle el número de opciones, los valores actuales, nid (número de la encuesta) y el texto de cada opción. Además para la comunicación entre la aplicación Flash y Drupal utilizamos una función javascript. Esta función será llamada desde la aplicación Flash pasando el valor elegido por el usuario codificado en MD5, para prevenir ataques de hackers, junto con el nid.

La función javascript llama sincronamente a la función voteit de advpoll a través de la url (<http://host/drupal/?q=advpoll/voteit>). Las variables se lo pasa en el POST.

La codificación del valor del voto es muy simple, se realiza un MD5 del string formado por el nid y el valor que puede valer entre 1 o 5. Teniendo el nid y la cadena codificada es muy sencillo obtener el valor elegido por el usuario.

Después llamamos a la función advpoll_vote para que guarde en la base de datos el voto generado anteriormente. Éste llama a los API'S VotingApi para realizar la acción y finalmente recalcula los votos totales que tiene la encuesta y lo devuelve a la función javascript desde donde le llamó. La aplicación Flash al llamar a la función se queda en un estado ocioso hasta que este le devuelva los totales o algún error. Si todo ha ido correctamente, la aplicación Flash genera un texto notificando al usuario que la acción ha finalizado correctamente.

Votar comentario

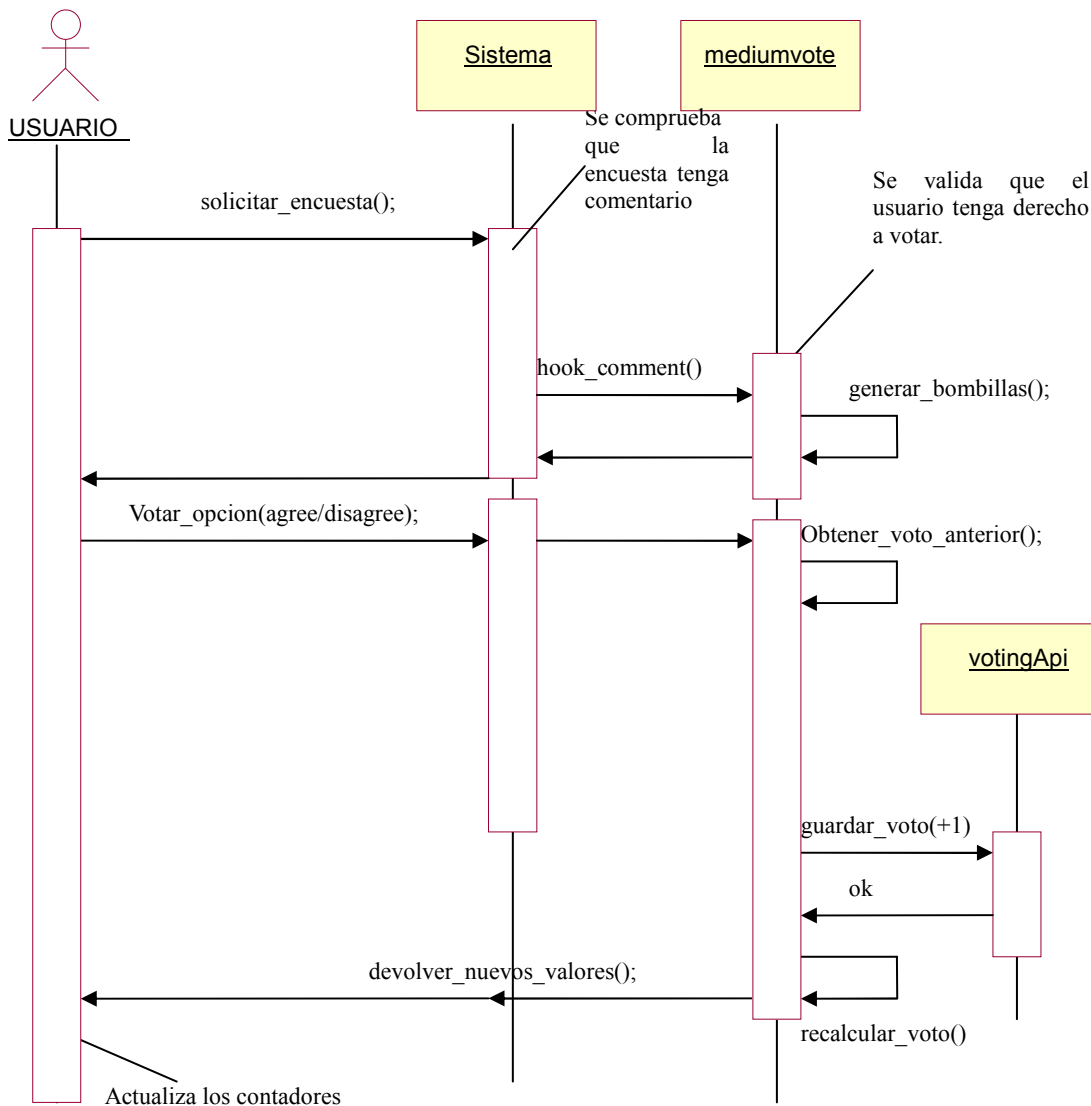


Figura: 5-8 Diagramas de secuencia. Votar comentario

Implementación en Drupal

Para implementar esta funcionalidad se ha tenido que crear a partir de un módulo base llamado Mediumvote. El proceso es muy similar al anterior, al solicitar una encuesta, el sistema determina si existe comentario. Si existiera el sistema llama al hook_comment. Al ser llamado nuestro modulo, se verifica que el usuario pueda votar, para ello se va a la tabla **votingapi_vote** y se comprueba que el usuario con uid no esté en la tabla. Se comprueba que el tiempo entre el último voto y el tiempo actual haya una diferencia de 1 hora. Llegamos a esta conclusión para prevenir el voto reiterado de algún usuario. Así prevenimos los votos fraudulentos. Si el usuario tiene permiso para votar se genera un tag html "span" en el cuál en el campo atributo lleva la url del voto. Ese atributo se cambiará por un evento onclick del ratón cuando se cargue la página, la función Javascript **mediumvoteClassAutoAttach** recorre todos los span de la página que tenga la clase vote_pos o vote_neg. Al

encontrar uno obtiene la url y elimina el atributo para prevenir futuros usos indebidos. Con el url, el identificador del span y del padre de éste se crea un objeto con el evento onclick.

El usuario anónimo/registrado al leer el comentario y le a convencido decide darle un voto a favor. Al apretar la bombilla verde se genera un evento, mediante javascript se lanza una solicitud asincrona (AJAX) para tratar el voto. El sistema obtiene los parámetros y obtiene el valor del voto anterior si lo tuviera y se le suma un voto más. Este voto es actualizado o insertado si es la primera vez que vota sobre ese comentario, esto se logra con una llamada a la función **votingapi_set_vote** del módulo votingapi. Finalmente solo nos queda obtener los nuevos valores y devolver a la función javascript para que actualice el contenido de los contadores a través del DOM. Teniendo la id del padres se puede acceder a los hijos fácilmente.

5.7 Creación de módulos en Drupal

Existe una manera para proveer nuevas funcionalidades a Drupal creando módulos adicionales con la funcionalidad que se desea darle. En este apartado introduciremos un pequeño tutorial de cómo crear un módulo que nos ha servido para calcular las estadísticas de los votos de la web.

Primero de todo hay que crear dos ficheros (*modulo.install* y *modulo.module*) situado en un mismo directorio. El fichero modulo.install tiene la función de crear y destruir tablas necesarias para el correcto funcionamiento de nuestro modulo, se explico anteriormente y damos por entendido.

El fichero modulo.module es un fichero php con extensión module, según la nomenclatura de Drupal tiene asignado así.

El hook_menu define la asociación entre un URL y la función que crea el contenido para el URL. También verifica los permisos necesarios para poder llamar al URL. Veamos nuestro ejemplo:

```
function calculatevote_menu($may_cache) {

  $items = array();

  if ($may_cache) {
    $items[] = array(
      'path' => 'calculatevote/run',
      'title' => t('Calculate vote manual'),
      'callback' => 'calculatevote_run',
      'access' => user_access('administer calculatevote'),
      'type' => MENU_CALLBACK
    );
  }

  return $items;
}
```

Las funciones que son llamadas desde el url se guardan en la variable \$items que es un array de array que contiene el path, el titulo, la función encargado de recoger la llama, el acceso y el tipo. Para nosotros solamente hemos tenido

que crear una, es accesible desde <http://host/drupal/?=calculatevote/run> y si lo llama algún usuario con permiso de “administer calculatevote”, el sistema llamara a la función calculatevote_run.

En hook_help, muestra una descripción en la lista de módulos. Nosotros hemos puesto la descripción de su funcionalidad. Calcula los votos diarios, semanales y mensuales de los líderes de opinión y de las encuestas más votadas.

```
function calculatevote_help($section) {
  switch ($section) {
    case 'admin/modules#description':
      return t('Calculate vote dairy,week,month');
      break;
  }
}
```

Hook_settings permite al administrador cambiar la configuración del funcionamiento del módulo. Normalmente son variables de configuración y hay que guardarlo en la variable \$form para que luego pueda generar el formulario correspondiente. Para nuestra web necesitábamos saber el intervalo de ejecución del trabajo de recuento. Por defecto hemos puesto a 3600.

```
function calculatevote_settings() {

  $form['calculatevote_settings'] = array(
    '#type' => 'fieldset',
    '#title' => t('Calculatevote Settings'),
    '#collapsible' => TRUE,
    '#collapsed' => false,
  );

  $form['calculatevote_settings']['calculatevote_interval'] = array(
    '#type' => 'textfield',
    '#title' => t('Interval exe. cron task for vote'),
    '#default_value' => variable_get('calculatevote_interval', 3600),
    '#description' => t('Time in seconds what execute cron task for recalculate vote for dairy, week, month stadistic. 3600 equals 1 hour.'),
  );

  return ($form);
}
```

Hook_cron, se implementa funcionalidades periódicas. Por ejemplo reindexar la búsqueda cada cierto tiempo para mejorar la búsqueda. Otro ejemplo podría ser cerrar las encuestas ya caducadas. En nuestro ejemplo llamaremos a las distintas funciones para el recalcule de los votos.

Hook_perm, devuelve la lista de permisos existentes en el módulo actual.

También vimos dos hook en el apartado anterior el Hook_view y el Hook_comment. El hook_view se llama cuando quieres mostrar algún contenido en los nodos y en el caso de hook_comment en los comentarios.

Capítulo 6. Conclusión

Los objetivos que habían sido propuestos al comienzo del proyecto han sido cumplidos. Todas las funcionalidades que se pidió han estado cumplidas.

Durante la realización de este proyecto he conseguido aprender unos conocimientos elevado sobre la programación web que me pueden abrir nuevas puertas en el mundo laboral. Antes de comenzar el proyecto, tenía serias dudas para abordar este proyecto. En la universidad el poco código HTML que se ha visto perjudica seriamente a los informáticos. Es un campo que cualquier persona con un buen Manuel llega a conseguir lo mismo que un informático. Dejando las comparaciones a parte. Un CMS nos permite aplicar el principio de la reusabilidad de un software, según el comentario de un experto informático:

“Los buenos programadores saben programar y los mejores programadores son los que saben reutilizar código”

Drupal es una comunidad muy avanzada con más de 600 módulos disponible para ser bajada y instalado por cualquier persona. Solucionan muchos de los requerimientos que se plantea, para esta proyecto he tenido que profundizar mucho en el funcionamiento de Drupal. Drupal es como una herramienta genérica que sirve para todo, pero siempre tendrás que tocar algo de código para adaptarlo a tus necesidades.

Se ha definido todas las funcionalidades del sistema en el apartado análisis y luego se ha diseñado para poder implementarlo en Drupal. Consiguiendo las opciones de votar en las encuestas, proporcional un link dejado de las encuestas para enviar a un amigo.

Estando en la fase de instalación y implantación en el servidor web www.rankitt.com se espera conseguir reunir a la gente para debatir las encuestas.

Los usuarios actualmente ya dispone a su alcance decidir con sus votos el contenido de la página principal. Esté contenido es actualizado a la función cron que proporciona Drupal para programar tarea a ejecutar repetidamente.

Al crear los dos nuevos módulo se ha definido dos nuevas tablas junto con las tablas que venían en Drupal se ha definido un nuevo modelo de datos que nos permite solucionar las distintas tareas propuestas.

7. Bibliografía

K. VanDyk, John; Westgate, Matt. *Pro Drupal Development*. Apress. 2007

T. Douglass, Robert; Little, Mike; W.Smith, Jared. *Building Online Communities with Drupal, phpBB, and WordPress*. Apress. 2006

Mercer, David. *Drupal: Creating Blogs, Forums, Portals, and Community Websites*. Packt Publishing. 2006

Comparación Drupal, Joomla, Plone.

http://www.idealware.org/articles/joomla_drupal_plone.php

Drupal Handbooks. <http://drupal.org/handbooks>

Drupal API. <http://api.drupal.org/api/group/database/4.7>

Drupal. <http://drupal.org/download>

Módulos drupal. <http://drupal.org/project/Modules>

Themes Drupal. <http://drupal.org/project/Themes>

8. Glosario

Avatar: En la religión hindú, los *Avatars* son encarnaciones en la tierra de un dios. Actualmente en las nuevas tecnologías y en Internet, se asocia la palabra avatar a la representación gráfica (mediante un dibujo o fotografía) de una persona para su identificación.



Avatar por defecto de rankitt.com

Banear: restringir el acceso a un sistema. Se puede restringir por nombre de usuario o por IP.

Folksonomía: Calco del inglés folksonomy, es un neologismo que da nombre a la categorización colaborativa por medio de etiquetas simples en un espacio de nombres llano, sin jerarquías ni relaciones de parentesco predeterminadas. Se trata de una práctica que se produce en entornos de software social. A diferencia de otros sistemas de categorización, las etiquetas de las categorizaciones se comparten entre los usuarios. Las folksonomías surgen cuando varios usuarios colaboran en la descripción de un mismo material informativo.

Society cyborgs del.icio.us
Design Digital-physical
convergence Education
Emergence Encyclopedia of the 21st
Century Energy Entertainment
Environment Events Futures
Game Geospatial Web Global risks
Globalization IFTF
Information
technology Innovation
Interaction Marketing Materials

Ejemplo de folksonomía

iCal: iCal es una aplicación de calendario personal hecha por Apple Computer que se ejecuta en el sistema operativo Mac OS X. iCal también puede referirse a iCalendar, el formato de calendario estándar empleado por la aplicación de calendario iCal. Sus principales características son:

- Almacena seguimiento de eventos y citas, permite múltiples vistas de calendario.
- Está integrado con .Mac por lo que los calendarios pueden ser compartidos por internet.
- Se puede suscribir a otros calendarios para mantener el contacto con amigos y otras cosas como horarios de competiciones atléticas y programas de televisión.
- Permite la recepción de notificaciones de próximos eventos tanto en la

pantalla, por correo electrónico, SMS o buscapersonas.

RSS: RSS es parte de la familia de los formatos XML desarrollado específicamente para todo tipo de sitios que se actualicen con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas. A esto se le conoce como redifusión o sindicación web.



El siguiente icono señala la presencia de feeds RSS convertido en estándar de facto.

9. Índice Figuras

Figura 2.1	<i>"Mapa meme" de la web 2.0 realizada en una conferencia en O'Reilly Media</i>
Figura 2.2	<i>Esquema de un CMS</i>
Figura 2.3	<i>El mercado de los CMS para los bloggers según Google</i>
Figura 2.4	<i>Números de búsquedas realizadas sobre los distintos CMS</i>
Figura 2.5	<i>Cantidad de acceso a las páginas a los webs del proyecto</i>
Figura 2.6	<i>Propiedades de los distintos sistemas</i>
Figura 3.1	<i>Disciplinas de cada Iteración</i>
Figura 3.2	<i>Esfuerzo asociado a las disciplinas según la fase</i>
Figura 3.3	<i>Proceso de desarrollo a partir de los Casos de Uso</i>
Figura 3.4	<i>Proceso de desarrollo de Software Macro Nivel</i>
Figura 3.5	<i>Etapas Concepción y Elaboración</i>
Figura 3.6	<i>Etapa de Construcción.</i>
Figura 3.7	<i>Etapa de Construcción: Ciclos de desarrollo (I)</i>
Figura 3.8	<i>Etapa de Construcción: Ciclos de desarrollo (II)</i>
Figura 4.1	<i>Diagrama de casos de uso</i>
Figura 4.2	<i>Planificación temporal del proyecto</i>
Figura 5.1	<i>Estructura de base de datos predefinida de Drupal</i>
Figura 5.2	<i>Diseño de la tabla que se ha tenido que crear.</i>
Figura 5.3	<i>Cabecera de la web</i>
Figura 5.4	<i>Ejemplo de encuesta</i>
Figura 5.5	<i>Bombillas para votar en los comentarios</i>
Figura 5.6	<i>Diagramas de pantallas</i>
Figura 5.7	<i>Diagramas de secuencia. Votar encuesta</i>
Figura 5.8	<i>Diagramas de secuencia. Votar comentario</i>

Anexo I. Drupal

En este anexo vamos a explicar con más detalles sobre el funcionamiento de Drupal, detallaremos las tecnologías necesarias para poder hacer funcionar a este CMS. También hablaremos de su organización interna.

I.1 Inicio de Drupal



Drupal es el equivalente fonético en inglés a la palabra neerlandesa “druppel” que significa gota, por eso el icono del sitio <http://drupal.org> es una gota sonriente.

Drupal fue originalmente escrito por Dries Buytaert y es el software usado para impulsar los sitios web Debian Planet, Terminus 1525, Spread Firebox y Kernel Trap.

A pesar de que empezó como un pequeño BBS, Drupal ha llegado a ser mucho más que sólo un portal de noticias gracias a su arquitectura flexible. Drupal se compone de una infraestructura base y un conjunto de módulos que ofrecen un amplio conjunto de funciones, incluyendo sistemas de comercio electrónico, galerías de fotos, administración de listas de correo electrónico, e integración de CVS. Es posible añadir módulos de terceros para modificar el comportamiento de Drupal u ofrecer nuevas funciones.

Drupal se distribuye bajo la licencia GNU GLP, y por lo tanto es software libre. Según su creador ese paso fue la mas importante que dio y conocemos el Drupal de hoy es gracias a millones de personas que ayudan a que este proyecto siga creciendo.

I.2 Pila de Tecnología de Drupal

Drupal tiene como meta ser un sistema que se pueda ejecutar bien en sitios web de pocos recursos y también ser capaz de escalar a sitios distribuidos masivos. Se pretende adaptar a cualquier sistema. Para ello, Drupal necesita unas tecnologías para poder funcionar.

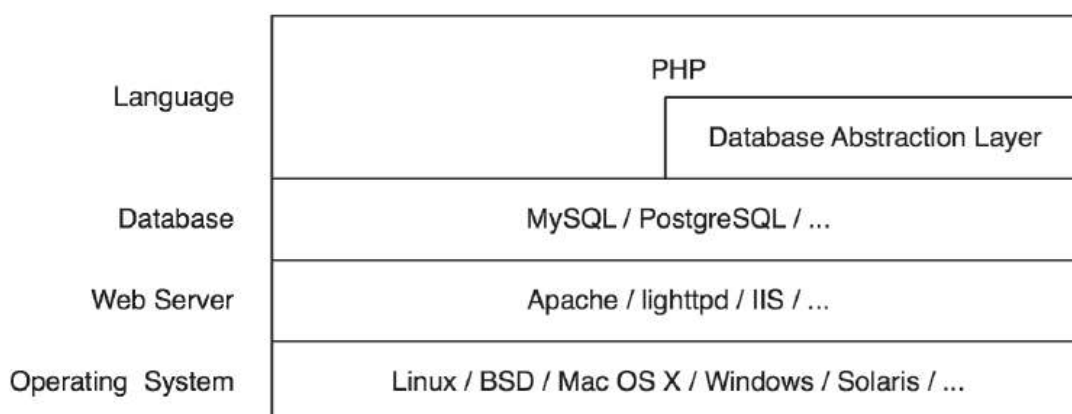


Figura: A-1 Tecnología base donde se sustenta Drupal.

Según se observa en la figura, el sistema operativo se encuentra en la base de la pila de tecnología, pero no hay que darle mucha importancia ya que Drupal puede funcionar en cualquier sistema operativo.

El servidor Web mas usado por Drupal es Apache, aunque otros servidores web también pueden ser usados (incluido Microsoft IIS). De esa estrecha relación, Drupal se ha adaptado en su funcionamiento, proviniendo archivo específico de configuración para ello (.htaccess) y utilización del modulo mod_rewrite para tener unos URL limpios sin marcas, ampersands o otros caracteres extraños. Por lo tanto si queremos utilizar otro servidor web tendremos adaptar el fichero de configuración .htaccess a una nueva sintaxis que el servidor pueda entender y también el modulo mod_rewrite con su homologo en el nuevo servidor.

Los sistemas de base de datos mas utilizado son MySQL y PostgreSQL, para ello la interfaz drupal para comunicarse con esta capa ha creado una capa abstracta. Así cualquier otra base de datos seria posible de utilizar añadiendo las funciones necesarias.

Finalmente la ultima capa se encuentra el lenguaje que esta hecho el Drupal. Todo el sistema esta codificado en PHP siguiendo estrictamente el estándar de codificación.

I.3 Núcleo de Drupal

El núcleo es la parte más importante y tiene la responsabilidad de proveer las funciones mas básicas que son usados para dar soporte a otras partes del sistema.

El núcleo incluye código que permite a Drupal capturar una petición por parte del servidor, una librería común de funciones que es usado frecuentemente por Drupal, y módulos que provee funcionalidades básicas como gestión de usuario, taxonomia, y templates. En la figura se observa mejor lo que hemos comentado anteriormente.

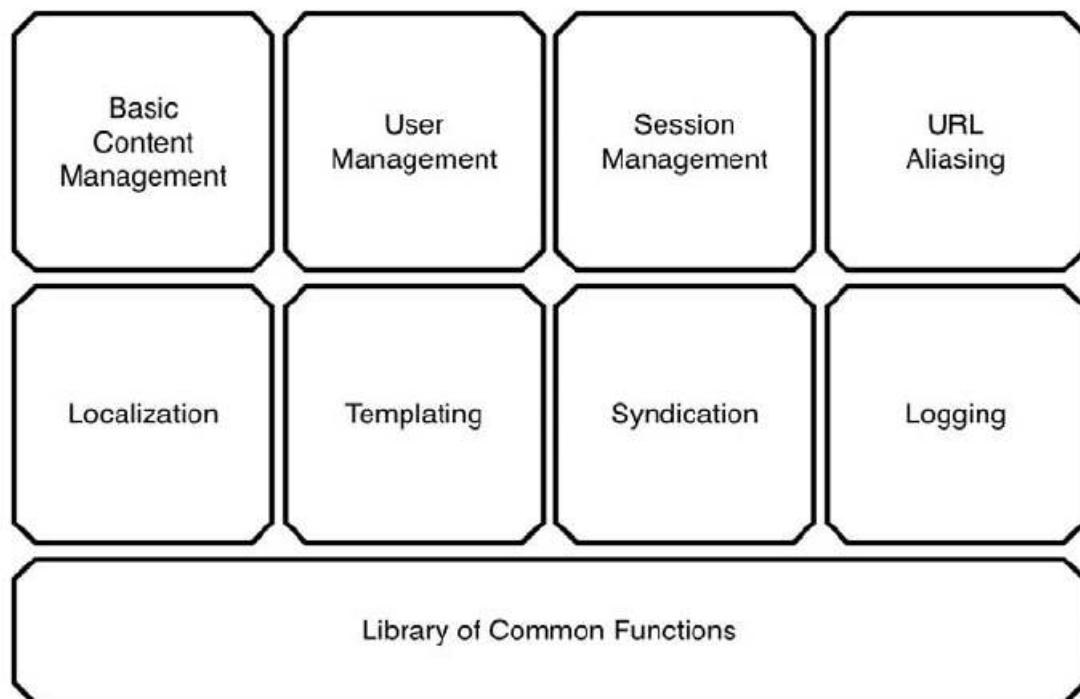


Figura: A-2 Núcleo de Drupal (Aparece las funcionalidades mas importantes)

I.4 Módulos

Drupal es verdaderamente un framework modular. Funcionalidad esta incluido en los módulos, los cuales se puede activar o desactivar (algunos no se pueden desactivar porque vienen por defecto junto con el núcleo y es necesario para su correcta funcionamiento). Añadiendo nuevos módulos se consigue nuevas funcionalidades para la web. Existe multitud de módulos creado por la comunidad Drupal que esta disponible en la página web del proyecto Drupal. Si llegase el caso que no encontramos esa nueva funcionalidad podemos nosotros mismo crear uno y luego compartirlo. Drupal hace uso del patrón de diseño de "inversion of control", en el cual las funcionalidades es llamado por el framework en el momento apropiado. Se consigue utilizando los hooks.

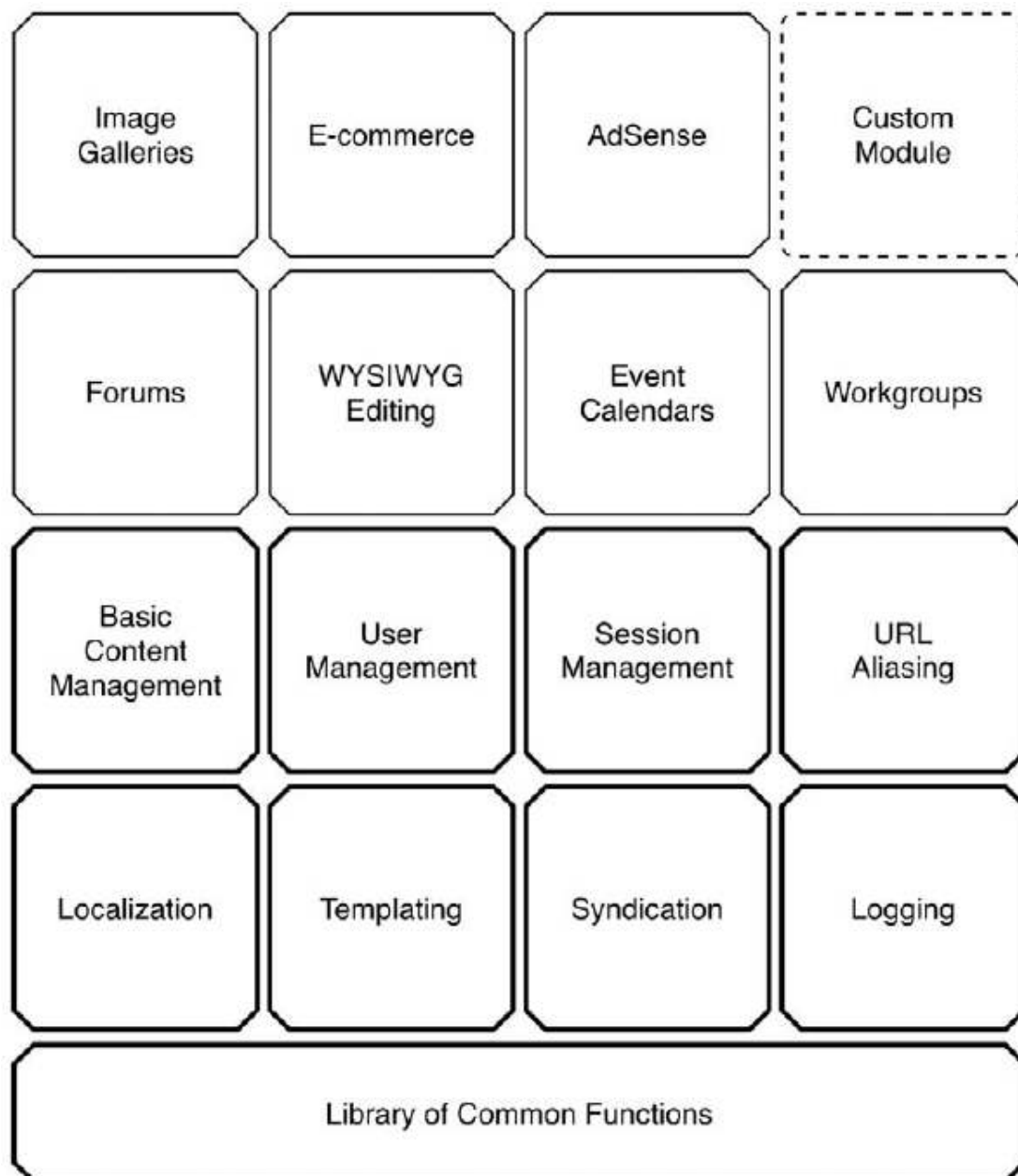


Figura: A-3 Activando módulos adicionales se consigue mas funcionalidades.

1.5 Hooks

Hooks puede definirse como eventos internos de Drupal, también llamado callbacks, se da este nombre porque en Drupal los hooks se implementan con una llamada a una función y no se necesita registrar ningún listener. Ya que drupal no esta implementado con objetos debido que el lenguaje nativo donde se basa drupal es el PHP. Cuando se inició el proyecto éste aún no incorporaba la posibilidad de programación orientado a objeto (OOP). Pero si se utiliza algunas propiedades de los OOP como objetos, abstracción, encapsulación, polimorfismo, herencia.

Hooks permite a los módulos saber lo que esta ocurriendo en todo el sistema.

Los hooks tiene la forma de `foo_bar()`, donde "foo" es el nombre del módulo

que implementa el evento y “bar” es el nombre del hook. Cada hook tiene predefinido un conjunto de parámetros y un retorno específico.

Para extender Drupal, un modulo necesita simplemente implementar un hook.

Para entender mejor vamos a poner un ejemplo: suponemos que el usuario hace un login en nuestra web. En el momento que el usuario entra, el sistema provoca un hook del tipo user. Esto significa que cualquier función que tenga implementado el hook user será llamado. `Comment_user()` en el módulo comentario, `locale_user()` en el módulo locale, etc...

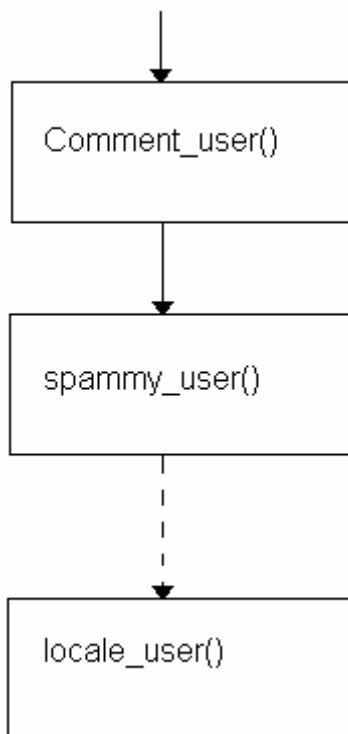


Figura: A-4 Flujo de ejecución de un hook.

I.6 Nodos

Los tipos de contenidos derivan del tipo base que se llama node. Una entrada de blog, un recipiente o un artículo todo tienen la misma estructura de dato. Los desarrolladores de módulos pueden añadir nuevas características como comentarios, adjuntar ficheros, información geográfica sin importar el tipo de nodo que sea.

Cualquier nodo puede ser candidato para ser mostrado en la página principal.

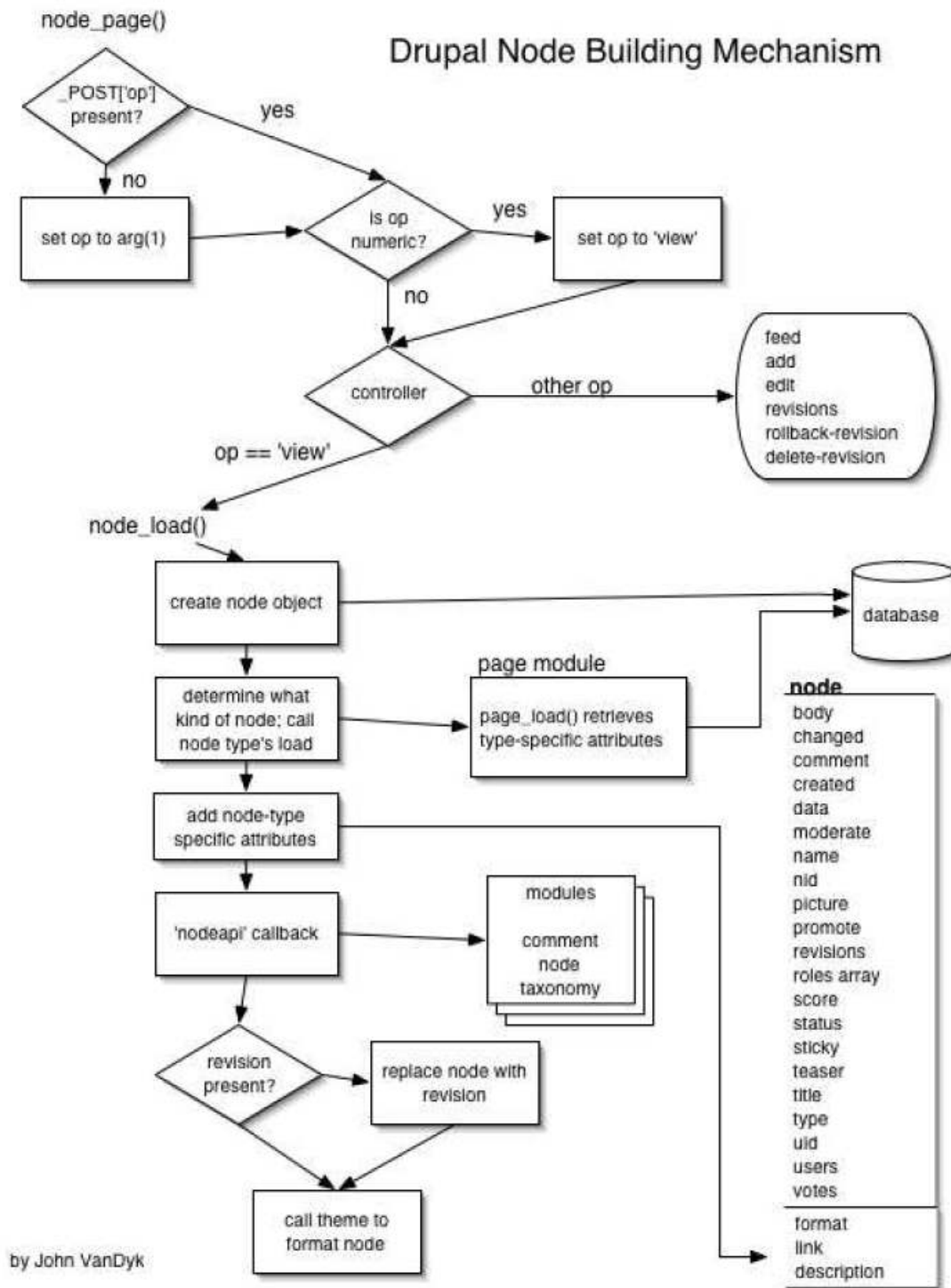


Figura: A-5 Mecanismo de creación de nodos.

El mecanismo de creación de nodos parte cuando el controlador chequea la `$_POST['op']` donde viene la operación que se va a hacer con el nodo. Cargamos la base del nodo accediendo a la base de datos. Dependiendo del tipo de nodo que sea se llama a su correspondiente función de load para añadir más datos al objeto base node. A continuación los módulos que tenga implementado el hook también serán accedido y finalmente se llama a la función theme para sacar el código html que se mostrará en el navegador.

I.7 Blocks

Los bloques son informaciones que se colocan los laterales, en la cabecera o en la parte inferior de nuestra página web. Estos pueden ser activado o desactivado para determinados tipos de nodos, o solamente presente en la página principal, o según otro criterio. Por ejemplo queremos mostrar una página distinta para cada tipo de usuario, en este caso usaríamos los bloques.

I.8 Themes

La capa de Theme es el responsable de crear el HTML que el navegador recibirá. Drupal utiliza populares templates para ellos como Smarty Template Attribute Language for PHP (PHPTAL) y PHPTemplate. Drupal permite varias maneras de crear el “look and feel” de nuestra página web. The método más simple es utilizar CSS para las clases y Id predefinido por Drupal. Otra manera consiste adaptar esos template a la estética que quiera tener nuestra página web. Las partes dinámicas (como box, listas, ...) pueden ser sobrescrito simplemente declarando una función con un nombre apropiado. En ese caso Drupal usará tu función en vez del suyo.

I.9 Estructura de directorio.

Entender la estructura de directorios nos pueden ayudar a depurar y sabremos donde hay que ir los nuevos módulos que nos bajemos o creemos. Una instalación por defecto nos proporciona la siguiente estructura:

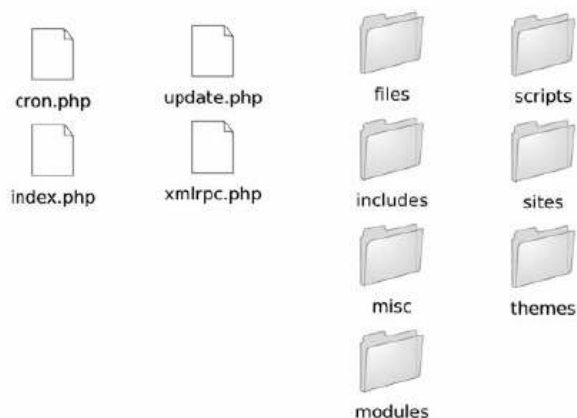


Figura: A-6 Estructura de directorio por defecto de Drupal.

files: en este directorio van el logo de la web, los ficheros de los avatares de los usuarios o otros ficheros multimedia que suba los usuarios.

- **includes:** contienen las librerías de las funciones más comunes del sistema.
- **misc:** aquí van los ficheros JavaScript, iconos e imágenes que son necesarios para que funcionen Drupal.
- **modules:** directorio que contiene los módulos del núcleo, además aquí irán todos los nuevos módulos que se creen o se añada. Drupal busca en este directorio todos los módulos que pueden ser activado y ser utilizado por el sistema.
- **scripts:** contienen script para analizar el sintaxis. Utilizan script creado en Perl.
- **sites:** contiene el fichero de configuración de nuestro sitio, debe de indicar el usuario, password y nombre de la base de datos que se va a utilizar.
- **themes:** contiene todos los temas que se admiten en nuestra web.

También se puede observar los distintos ficheros php que hay. El index.php procesa todas las peticiones que vaya a nuestra web. El fichero cron.php sirve para ejecutar tareas periódicas, en nuestro caso hemos hecho servir este proceso para recalcular los usuarios con más votos, las encuestas y ranking más votados durante el día, semana y mes. Así conseguimos que el servidor no haya que ir a calcularlo cada vez que un usuario vea una página web. Con lo que conseguimos que la web vaya más rápido. El fichero xmlrpc.php sirve para procesar las peticiones de llamadas remotas de procedimientos. Y finalmente esta el fichero update.php que actualiza nuestra base de datos cuando añadimos o quitamos módulos, ya que estos crean tablas para trabajar.

I.10 Procesando una petición al servidor.

Al introducir un usuario una url de nuestra web drupal. El fichero index.php incluye la include bootstrap.inc que ejecuta unos pasos cada vez es llamado:

- **Configuration:** carga la configuración necesaria como el URL base y las variables de configuración del sistema.
- **Database:** durante este paso, el sistema determina el tipo de base de datos (MySQL o PosgreSQL, los más utilizado) y inicializa la conexión con la base de datos determinado.
- **Access:** Drupal permite banear (no permitir el acceso) de un determinado host. En esta fase se verifica que el host no fue baneado, sino mostramos la página de acceso denegado.
- **Session:** en esta fase se inicializa la sesión del usuario o si ya fue creado lo recupera.
- **Late Page Cache:** determina si la cache esta activado si la página web que solicita esta en el cache. Si es afirmativo devuelve la página guardado en la cache. Sino sigue el proceso.
- **Path:** Se carga el módulo de alias de path y calcula la path exacta que se pasará al sistema.
- **Full:** Carga las librerías más comunes.

Llegado a este paso, se inicializa los módulos llamando a la función `module_init()`. Dentro de esta función se obtiene la lista de módulos que se va a llamar. Después de obtener la lista se llama al hook `init` de todos esos módulos.

El siguiente paso es determinar si el usuario tiene permisos para realizar la acción. Si `user_access()` devuelve un 0 significa que el usuario no tiene suficiente permisos y por lo tanto le mostramos su correspondiente página.

Finalmente es llamado `init_theme()` proporcionando una estética a los datos devueltos cuando se llamó al `module_init()`.

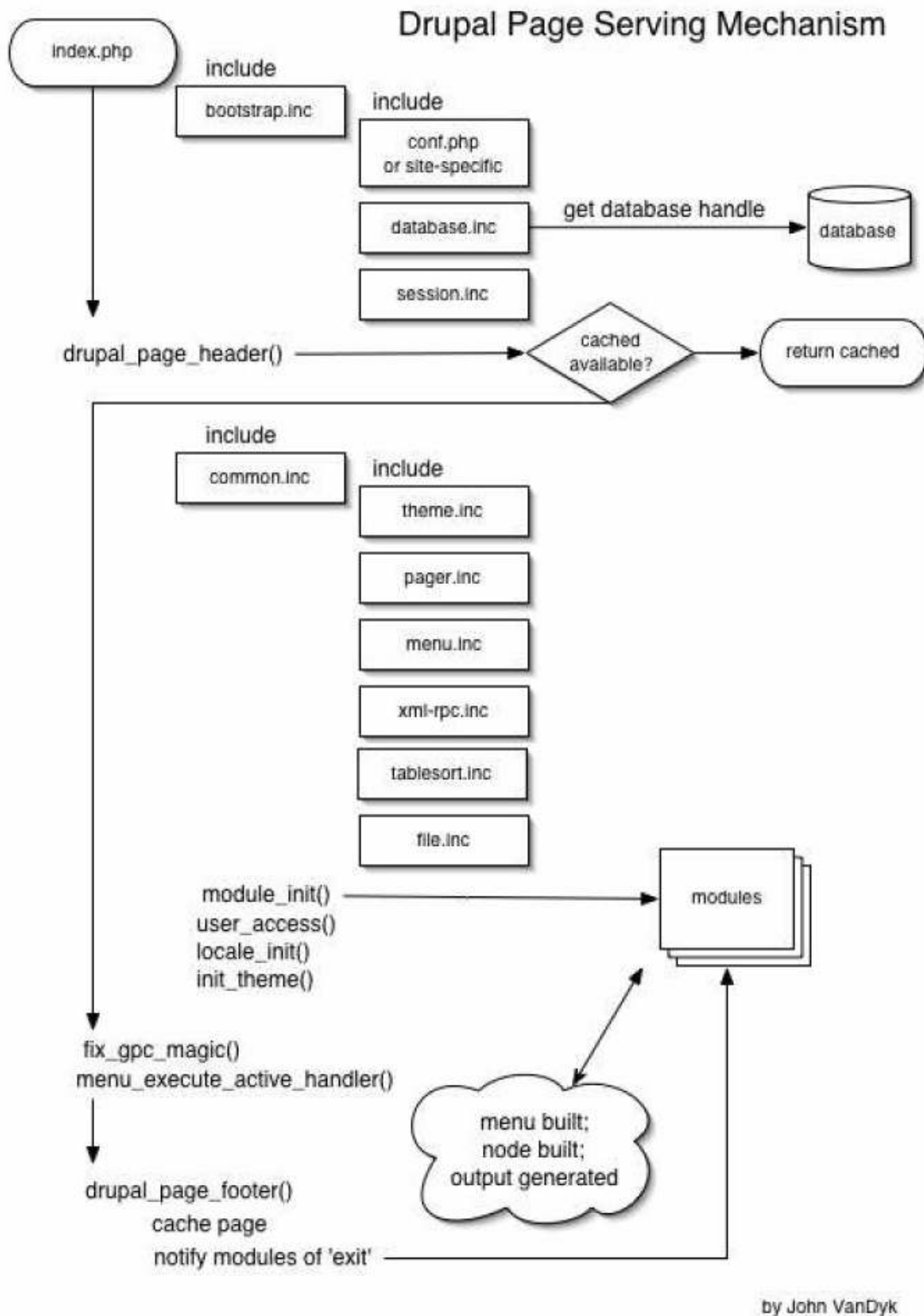


Figura: A-7 Mecanismo de servir paginas en Drupal.