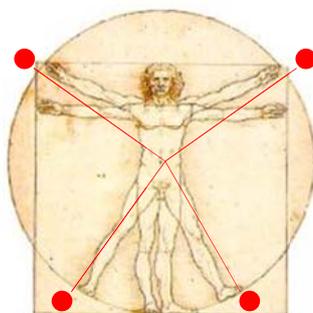


TECNOLOGÍ@ y *DESARROLLO*

Revista de Ciencia, Tecnología y Medio Ambiente

VOLUMEN XIV. AÑO 2016
SEPARATA



**SISTEMA DE GESTIÓN DE CONTENIDOS Y
TIENDA ONLINE SINCRONIZADA CON ERP**
Óscar Notario Cuadrado, Antonio J. Reinoso



UNIVERSIDAD ALFONSO X EL SABIO

Escuela Politécnica Superior
Villanueva de la Cañada (Madrid)

© Del texto: Óscar Notario Cuadrado, Antonio J. Reinoso
Abril, 2016.

<http://www.uax.es/publicacion/sistema-de-gestion-de-contenidos-y-tienda-online-sincronizada-con-erp.pdf>

© De la edición: Revista Tecnológí@ y desarrollo
Escuela Politécnica Superior.
Universidad Alfonso X el Sabio.
28691, Villanueva de la Cañada (Madrid).
ISSN: 1696-8085
Editor: Javier Morales Pérez – tecnologia@uax.es

No está permitida la reproducción total o parcial de este artículo, ni su almacenamiento o transmisión ya sea electrónico, químico, mecánico, por fotocopia u otros métodos, sin permiso previo por escrito de la revista.

SISTEMA DE GESTIÓN DE CONTENIDOS Y TIENDA ONLINE SINCRONIZADA CON ERP

Óscar Notario Cuadrado, Antonio J. Reinoso^b

^{a)} Grado en Ingeniería de Sistemas de Información, Universidad Alfonso X el Sabio.
Avda. De la Universidad nº1, Villanueva de la Cañada, 28691, Madrid. España. onotacua@myuax.com

^{b)} Doctor Ingeniero en Informática, Adjunto a la Jefatura de Estudios
Departamento de Ingenierías TIC, Escuela Politécnica Superior, Universidad Alfonso X el Sabio.
Avda. De la Universidad nº1, Villanueva de la Cañada, 28691, Madrid. España. areinpei@myuax.com

RESUMEN Un sistema de gestión de contenidos (*CMS*) es un programa informático que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web. Esta publicación consiste en el desarrollo de un gestor de contenidos propio para la creación de páginas web a medida, mediante *Apache, PHP, MySQL, HTML, CSS, JavaScript, jQuery, Ajax y XML*.

Inicialmente se realizará un estudio de viabilidad para valorar las principales alternativas de gestores de contenido existentes en el mercado, tanto de código abierto como propietarios, y se llegará a la conclusión de que ninguna se ajusta a lo que realmente se quiere desarrollar.

Para el desarrollo de este gestor de contenidos web, primero se realizará un análisis para obtener una especificación detallada del sistema que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema, en el cual se definirá la arquitectura y los componentes del mismo.

Posteriormente se desarrollan todos los procedimientos de operación y seguridad, y se realizarán las pruebas unitarias, de integración y del sistema. Y por último, se describirá el procedimiento de implantación y aceptación del sistema.

PALABRAS CLAVE: Gestor de Contenidos, Web, CMS, Tienda online, Análisis, Diseño, Programación, Apache, PHP, mySQL

1. Introducción

Un sistema de gestión de contenidos, en inglés: *Content Management System* más conocido por sus siglas *CMS*, es un programa informático que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás usuarios.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio web. 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 web sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores (Varios, 2013-15).

Esto permite gestionar la información del servidor, reduciendo el tamaño de las páginas para descarga y reduciendo el coste de gestión del portal con respecto a un sitio web estático, en el que cada cambio de diseño debe ser realizado en todas las páginas

web, de la misma forma que cada vez que se agrega contenido tiene que maquetarse una nueva página HTML y subirla al servidor web.

El gestor de contenidos facilita el acceso a la publicación de contenidos a un rango mayor de usuarios. Permite que, sin conocimientos de programación ni maquetación, cualquier usuario pueda añadir contenido en el portal web.

Este artículo consiste en el análisis, diseño y programación de un gestor de contenidos propio y modular, que permita crear y editar contenidos de sitios web de manera sencilla. El desarrollo se realizará sobre Apache, PHP y mySQL, y estará basado en código abierto bajo licencia GPL.

2. Estudio de Alternativas CMS

Las clasificaciones de los Gestores de Contenidos comúnmente admitidas parten de tres elementos que permiten segmentarlos según diferentes criterios:

- *Lenguaje de programación o tecnología utilizada:*
 - Active Server Pages (ASP)
 - Java
 - PHP
 - ASP.NET
 - Ruby On Rails
 - Python
- *Funcionalidades que ofrece la aplicación:*
 - Plataformas generales Web
 - Sistemas específicos
 - Orientados a páginas personales: Blogs
 - Orientados a compartir opiniones: Foros
 - Orientados al desarrollo colaborativo: Wikis
 - Plataforma para contenidos de enseñanza on-line: e-learning
 - Plataformas de comercio electrónico o gestión de catálogos: ecommerce
 - Publicaciones digitales
 - Difusión de contenido multimedia
- *Propiedad del código:*
 - Código abierto (*Open Source*) o Software libre: que permite que se desarrolle sobre el código fuente.
 - Código propietario (aplicaciones comerciales): sólo su desarrollador puede modificar la aplicación.
 - Software as a Service: una nueva forma de obtener el servicio sin adquirir el producto, sea éste comercial o de código abierto.

Atendiendo a esta última clasificación, vamos a agrupar los principales gestores de contenido existentes en el mercado en Código Abierto o Código Propietario

2.1 Gestores de Contenido de Código Abierto

Las soluciones de código abierto son aquellas que independientemente de que hayan sido desarrolladas por una compañía o por una comunidad de usuarios tienen características comunes, tales como acceso al código fuente, posibilidad de redistribución de la aplicación y la posibilidad de adaptar el código a necesidades específicas. Existen multitud de términos y licencias englobadas en este concepto (por ejemplo, LGPL, GPL, Apache, MIT, Creative Commons) y con características diferentes pero en este informe por motivos de generalidad se usarán los términos código abierto y software libre como sinónimos que incluyen a todas esas licencias. (García-Cuerva, 2014)

Ventajas:

- Bajo coste de entrada (sin licencias)
- Dado que el código es abierto, las oportunidades para particularizar y añadir nuevas funcionalidades son mayores.
- También se pueden simplificar otros procesos como, por ejemplo, contratar diferentes proveedores servicio para realizar las modificaciones a lo largo del tiempo ya que se mantiene la propiedad del código.
- Es posible encontrar aplicaciones que se adapten directamente a las necesidades del negocio (ya sea directamente o mediante la inclusión de módulos adicionales) o bien que puedan integrarse varias aplicaciones para conseguir la funcionalidad deseada.
- Son más reactivos a cambios en las necesidades de los usuarios o a la adopción de nuevos estándares. Al ser modelos de desarrollo en comunidad cualquier desarrollador interesado puede crear un módulo que proporcione la nueva funcionalidad sin que exista un complejo proceso de toma de decisión.

Inconvenientes:

- Una de las principales desventajas es la incertidumbre acerca de la solución. Habitualmente, aspectos como el tiempo de vida de la solución, documentación, formación, solución de fallos en la aplicación, etc., dependen de los voluntarios que están involucrados en la comunidad de desarrollo. Como resultado, el tiempo necesario para poner en marcha la solución puede ser mayor que para una solución comercial.
- Puede haber problemas a la hora de realizar cambios o añadir nuevas funcionalidades, por no conocer al detalle cómo está desarrollado o por problemas con futuras actualizaciones aportadas por la comunidad.
- Al estar el código fuente disponible en internet, son más vulnerables a ataques de hacker o cracker, porque pueden conocer sus agujeros de seguridad.
- Al tratarse de soluciones genéricas, sobran muchas funcionalidades, por lo que la facilidad de uso y la velocidad de carga son menores.

El número de aplicaciones CMS de código abierto es enorme, lo que impide un análisis exhaustivo. Por ejemplo, CMS Matrix proporciona un servicio de comparativa de características de gestores de contenidos de código abierto, donde el número actual de soluciones indexadas es de alrededor de 1.200 soluciones. No obstante la fiabilidad, vitalidad y madurez de dichas soluciones varían enormemente.

Para nuestro análisis, hemos incluido productos y soluciones que cumplen uno o varios de los siguientes criterios:

- *Cuota de mercado:* Se consideran soluciones que tengan una cierta cuota de mercado y que, por tanto, hayan probado su eficacia en el uso real. También se pueden considerar productos que debido a la novedad en su enfoque o tecnología no hayan tenido tiempo de alcanzar una determinada cuota de mercado.
- *Vitalidad:* Se consideran productos estables y en continuo proceso de mejora. En un mundo tan cambiante como es el de las tecnologías Web, y más en código abierto, una forma de disminuir el riesgo es utilizar un producto que tenga una comunidad activa de usuarios y desarrolladores.
- *Tipo de producto:* Se consideran productos de distintas categorías y funcionalidades que incluyan soluciones para abordar la gran diversidad de necesidades planteadas por los clientes. Esto implica considerar desde productos muy simples hasta productos más completos que requieran mayores conocimientos y recursos.
- *Tecnologías:* Se adopta un enfoque neutral frente a la tecnología de implementación y se trata de incluir productos que usen las tecnologías Web más empleadas. En concreto, debe haber por lo menos algún producto con las siguientes tecnologías, PHP, Java y .NET y, por razones de cobertura, se puede considerar algún otro producto con tecnologías menos utilizadas (por ejemplo, Python).

A partir de estos criterios, las soluciones de código abierto inicialmente consideradas son:

- *Joomla!:* Gestor de contenidos Web con un gran número de funcionalidades que permitan cubrir las operaciones más habituales de una empresa.
- *WordPress:* Uno de los gestores de contenidos Web más usados, que surgió a partir de una iniciativa para crear un sistema de publicación personal en la Web (es decir una bitácora o blog), y se ha convertido en un completo CMS.
- *Drupal:* Gestor de contenidos Web que en su distribución estándar cubre prácticamente todos los aspectos de la gestión de información en la Web.
- *Mambo:* Sistema de portales CMS basado en el lenguaje de programación PHP y base de datos SQL de código abierto. Basa todo su aspecto en plantillas o temas.
- *Typo3:* Uno de los gestores de contenidos Web más antiguos que se basan en el desarrollo por una comunidad y tiene una amplia base de sitios Web de muy distintos sectores que lo utilizan.

- *CMS Made Simple*: Gestor muy simple de contenidos Web que permite crear un sitio Web de forma rápida.
- *Alfresco Community*: Tiene un modelo de código abierto y una versión empresarial que, aunque está basada en código abierto, tiene una suscripción para cubrir el soporte y el mantenimiento.
- *DotNetNuke*: Gestor de contenidos Web desarrollado con tecnología Microsoft .NET. Dispone de una versión gratuita pero que requiere disponer de la infraestructura de tecnología Microsoft, y de una versión con soporte comercial.
- *OpenCMS*: Desarrollado por la compañía Alkacon Software que también ofrece un servicio de soporte profesional. Es un producto simple de utilizar, pero incorpora aspectos sofisticados, como gestión de flujos de trabajo. Está basado en tecnologías Java y XML.
- *Plone*: CMS basado en el lenguaje de programación Python y que se ejecuta con el servidor de aplicaciones Zope.

2.2 Gestores de Contenido de Código Propietario

Las aplicaciones o soluciones de código propietario son aquellas que dependen directamente de una empresa determinada, que tiene la propiedad del producto y se ocupa de proporcionar los servicios más habituales de soporte, formación, mantenimiento, etc.

Ventajas:

- Productos generalmente más estables y normalmente con un compromiso de solución de problemas en plazos determinados.
- Ofrecen características específicas que no es habitual encontrarse en las soluciones de código abierto.
- Incluyen una completa documentación y se puede contratar formación respecto al producto.
- La solución propietaria puede tener incluidas directamente (o se pueden configurar) el conjunto de características necesarias que se adecuan más al negocio.

Inconvenientes:

- Mayor coste inicial que habitualmente supone su implantación debido a que hay que pagar algún tipo de licencia.
- La empresa propietaria es la que controla el tipo de modificaciones o extensiones que se le pueden hacer al producto. En el caso de que soporten dichas modificaciones, habitualmente tienen que ser desarrolladas por el proveedor original de la solución, de modo que el coste de la solución propietaria puede dispararse aún más.

Los principales gestores de contenido web de código propietario son los siguientes:

- *Autonomy*: Una de las empresas de software más grandes de Europa, y dispone del CMS Interwoven, que tiene una gestión integrada y aspectos de publicación Web.
- *EMC*: Tienen productos que abordan todos los aspectos claves de la gestión documental, lo que unido a su fuerte presencia en determinados mercados, hace que tenga una posición privilegiada en la gestión de contenidos web.
- *Google Sites*: Es un servicio ofrecido por Google Inc. que permite crear un sitio web mediante un editor sencillo).
- *Hyland*: Esta compañía con su producto On Base tiene una clara orientación estratégica que se basa en las tecnologías Microsoft y en la arquitectura .NET.
- *IBM*: Tiene diversos productos de gestión de contenidos como, por ejemplo FileNet o Lotus Web Content Management, que cubren todos los aspectos del proceso.
- *Microsoft*: Ofrece dos soluciones completamente distintas que cubren desde una oferta muy simple y gratuita con Office Live Small Business, similar al Google Sites, hasta una oferta profesional con Microsoft Office SharePoint Server 2010.
- *Oracle*: Una de las principales ventajas es que aunque hay competidores con soluciones CMS más maduras en algunos aspectos, se integra perfectamente con otros productos de la compañía.
- *Xerox*: Su producto DocuShare tiene una clara funcionalidad de colaboración y de integración sencilla de los documentos en la gestión de contenidos Web. Pero es sobre todo con su versión DocuShare Express, que tiene un precio mucho menor con la que está tratando de ganar cuota de mercado en las pequeñas y medianas empresas.

3. Valoración de las Alternativas

El proceso para la selección de una aplicación de gestión de contenidos para la realización de páginas web, o la realización de un gestor de contenidos propio es un proceso arduo y complejo, dada la gran cantidad de soluciones existentes en el mercado y los aspectos que deben considerarse para la toma de una decisión debidamente fundada (Consultrans, 2014).

Para facilitar la selección de un determinado gestor de contenidos o la realización de uno a medida, habría que estudiar tres niveles de criterios:

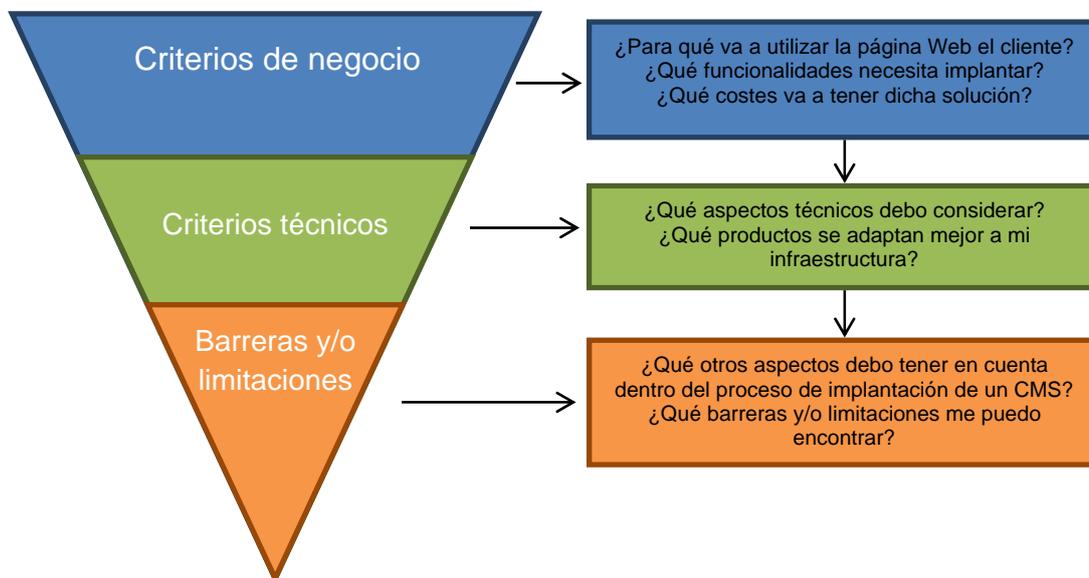


Figura 1. Criterios para la valoración de alternativas

3.1 Criterios de Negocio

Los criterios de negocio tienen como objetivo acotar el alcance de la estrategia en la red de una empresa. Estos criterios obedecen a la finalidad de establecer un marco a partir del cual se puede limitar el espectro de soluciones disponibles teniendo en cuenta las necesidades de la empresa en cuanto a su presencia en Internet.

Evidentemente no es lo mismo una empresa que quiera desarrollar un portal para tener presencia Web y/o mejorar sus relaciones con los empleados que una compañía que pretenda comercializar sus productos a través de su Web corporativa. Es muy importante que el cliente defina y acote con la mayor precisión posible, el tipo de servicio o funcionalidad que necesita.

Así mismo, el otro aspecto que deben tener en cuenta los criterios de negocio son los relativos a los costes económicos asociados a la implantación de una solución de gestión de contenidos. Estos costes deben contemplar todos los aspectos posibles (adquisición, implantación, mantenimiento, etc.), haciendo por tanto hincapié en el concepto de *coste total de propiedad* o TCO.

3.2 Criterios Técnicos

Los sistemas de gestión de contenidos Web son aplicaciones complejas y diversas, que incluyen capacidades muy diferentes y se basan en tecnologías muy variadas. Esto hace que la comparación general de productos sea muy difícil. A continuación se describen algunos de los criterios técnicos más relevantes:

- Tipo de licencia.
- Coste de implantación y mantenimiento.
- Infraestructura necesaria.
- Funcionalidades y extensibilidad.
- Simplicidad de uso.
- Soporte profesional, documentación y formación.
- Soporte de estándares.
- Estabilidad del producto.
- Soporte multilingüe.
- Indexación por buscadores.

Normalmente, todos estos aspectos tienen una influencia directa sobre los costes económicos asociados a la compra, adquisición, puesta en marcha, explotación y operación de cualquier sistema, aplicación o portal Web, y por ello se suelen agrupar bajo un criterio general denominado coste total de propiedad o TCO.

3.3 Barreras a la adopción e implantación de los CMS

Una vez analizados y ponderados los criterios anteriores, es importante que tengamos en cuenta los siguientes aspectos de cara a la decisión final sobre la herramienta o solución que mejor responde a nuestras:

- Conocimiento del mercado de soluciones.
- Búsqueda de proveedores adecuados.
- Tiempo de implantación.
- Mantenimiento y actualización de contenidos.
- Coste total de propiedad (implantación + mantenimiento).
- Relación coste-beneficio.

3.4 Requisitos Técnicos para el Alojamiento y Operación de Gestores de Contenidos

Los gestores de contenido Web, tanto de código abierto (*software libre*) como comerciales (*propietarios*), requieren para su instalación y operación de una infraestructura técnica para poder llevar a cabo su tarea, es decir, requieren de un servidor que cumpla unos requisitos.

Es importante no sólo tener en cuenta el coste asociado a la adquisición del propio gestor de contenidos (cuando se trate de un producto comercial), sino también el coste asociado a los requisitos técnicos e infraestructura necesaria para el alojamiento y explotación del mismo, ya que pueden requerir por ejemplo, que se adquieran licencias de otros productos como paso previo a la puesta en funcionamiento del gestor.

Aunque estos requisitos dependen del producto específico, existen cuatro grandes categorías de requisitos técnicos a considerar:

- *Plataforma de desarrollo*: El desarrollo de aplicaciones se lleva a cabo en base a una plataforma de desarrollo específica, vinculada a un determinado lenguaje de programación o tecnología (Por ejemplo: *PHP*, *.NET*, *Java*).
- *Sistema Operativo*: Es común que algunos productos, en particular los propietarios, requieran un Sistema Operativo específico (Por ejemplo: *Windows*, *Linux*) para poder instalar y poner en funcionamiento el gestor de contenidos.
- *Servidor web*: Los gestores de contenidos Web requieren de una aplicación encargada de gestionar la comunicación entre el servidor y los navegadores de los clientes (Por ejemplo: *Apache*, *Tomcat*, *IIS*).
- *Servidor de Base de Datos*: La información asociada al gestor de contenido es almacenada en un servidor de base de datos (Por ejemplo: *mySQL*, *Oracle*, *Postgress*). Un servidor de Base de Datos actúa como un almacén de información encargado de gestionar principalmente la consistencia de la información y la durabilidad de la misma.

Pese a que para el funcionamiento de cada gestor de contenido Web existen unos requisitos técnicos específicos, se suelen ofrecer múltiples alternativas para cada una de estas categorías, de modo que el usuario puede elegir (o ser asesorado) acerca de las ventajas y desventajas de cada una de las opciones posibles frente al coste de cada una de las soluciones específicas, aunque en ocasiones existe una fuerte vinculación a la hora de seleccionar los distintos productos específicos para cumplir con los requisitos técnicos, de modo que al seleccionar algún producto específico para cubrir un determinado requisito técnico, se restringen las posibilidades para el resto de requisitos, por ejemplo, en el caso particular del sistema operativo.

4. Selección de la Solución

Después de estudiar las diferentes alternativas en cuanto a gestores de contenidos web, y estudiar más en detalle varias de ellas, se ha optado por desarrollar un CMS propio porque, aunque inicialmente lleva mucho más trabajo su desarrollo íntegro, una vez terminado nos ofrecerá las siguientes ventajas:

No surgirán problemas en el momento de incorporar una nueva funcionalidad. Con cualquier otro gestor de contenidos, el cambio más simple puede convertirse en un grave problema si la solución no está desarrollada.

Se conocerá al detalle cómo está desarrollado y no habrá ningún problema a la hora de introducir cambios o actualizaciones, tanto en la programación de nuevas funcionalidades como en el diseño del sitio web.

La seguridad en un sitio web es de vital importancia. Un gestor de contenidos propio nos asegura que ningún hacker conoce el código fuente de las páginas web generadas.

La velocidad y estabilidad del sitio web es otro punto a tener muy en cuenta, porque si un usuario visita una página web con un tiempo de carga demasiado elevado, la

abandonará de inmediato y no volverá a visitarla. Además, los buscadores también penalizan a los sitios web muy lentos, afectando al posicionamiento en sus resultados de búsqueda.

Un gestor de contenidos a medida, podrá ser mucho más rápido gracias a que la base de datos y el código de programación están diseñados, desarrollados y optimizados para responder a los requerimientos del sitio web y sus directrices. No como los CMS comerciales, que son tan genéricos que introducen muchísimo código innecesario y eso ralentiza la ejecución de la página web.

Por último, otro de los motivos para desarrollar un gestor de contenidos propio es por la facilidad de uso tanto a nivel de los usuarios visitantes como del administrador de los contenidos. Será mucho más fácil de utilizar que los CMS comerciales, puesto que se podrá adaptar a las necesidades concretas de cada cliente.

5. Identificación de Subsistemas de Análisis

El sistema se descompondrá en los siguientes subsistemas o módulos, pudiendo ampliarse instalando módulos complementarios:

1. *Administración:* Panel de administración de todo el gestor de contenidos. Desde aquí se gestionará lo siguiente:
 - Secciones de la página web
 - Usuarios registrados y sus grupos
 - Permisos de los grupos de usuarios
 - Configuración general de la aplicación
2. *Álbumes:* Galerías de fotos. Las imágenes se pueden cargar de forma masiva y se optimizarán en 3 tamaños, uno de ellos con marca de agua. Una vez subidas se puede cambiar el pie de foto y ordenarlas arrastrando y soltando, y borrar.
3. *Contacto:* Envío de formularios de contacto desde la aplicación.
4. *Descargas:* Descargas de archivos. Los archivos se pueden cargar de forma masiva. Una vez subidos se puede cambiar su título, ordenar arrastrando y soltando, y borrar.
5. *Noticias:* Noticias con editor de texto enriquecido, imágenes y descargas.
6. *Páginas de Texto:* Página de texto con un editor enriquecido, imágenes y archivos.

Se utilizará también para guardar texto en varias demás secciones y para mostrar el mapa de localización, insertando el código HTML de Google Maps.
7. *Productos:* Catálogo de productos para la tienda online.
8. *Videos:* Galerías de videos.
9. *Pedidos:* Pedidos realizados desde la tienda online.

5.1 Especificación e Interfaces entre Subsistemas

Cada módulo será totalmente independiente de los demás, para poder instalarlos de forma totalmente transparente. Únicamente tendrá relación con el núcleo del sistema, que será obligatoria su instalación.

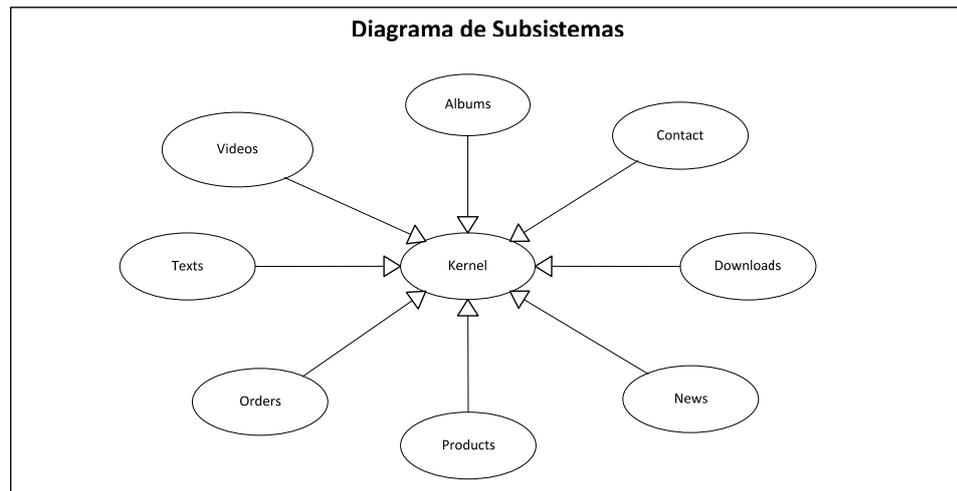


Figura -2. Diagrama de Subsistemas

6. Elaboración del Modelo de Datos

6.1 Elaboración del Modelo Conceptual de Datos: Entidad-Relación

En los siguientes diagramas se muestra el modelo conceptual de la Base de Datos, independientemente del gestor de base de datos utilizado.

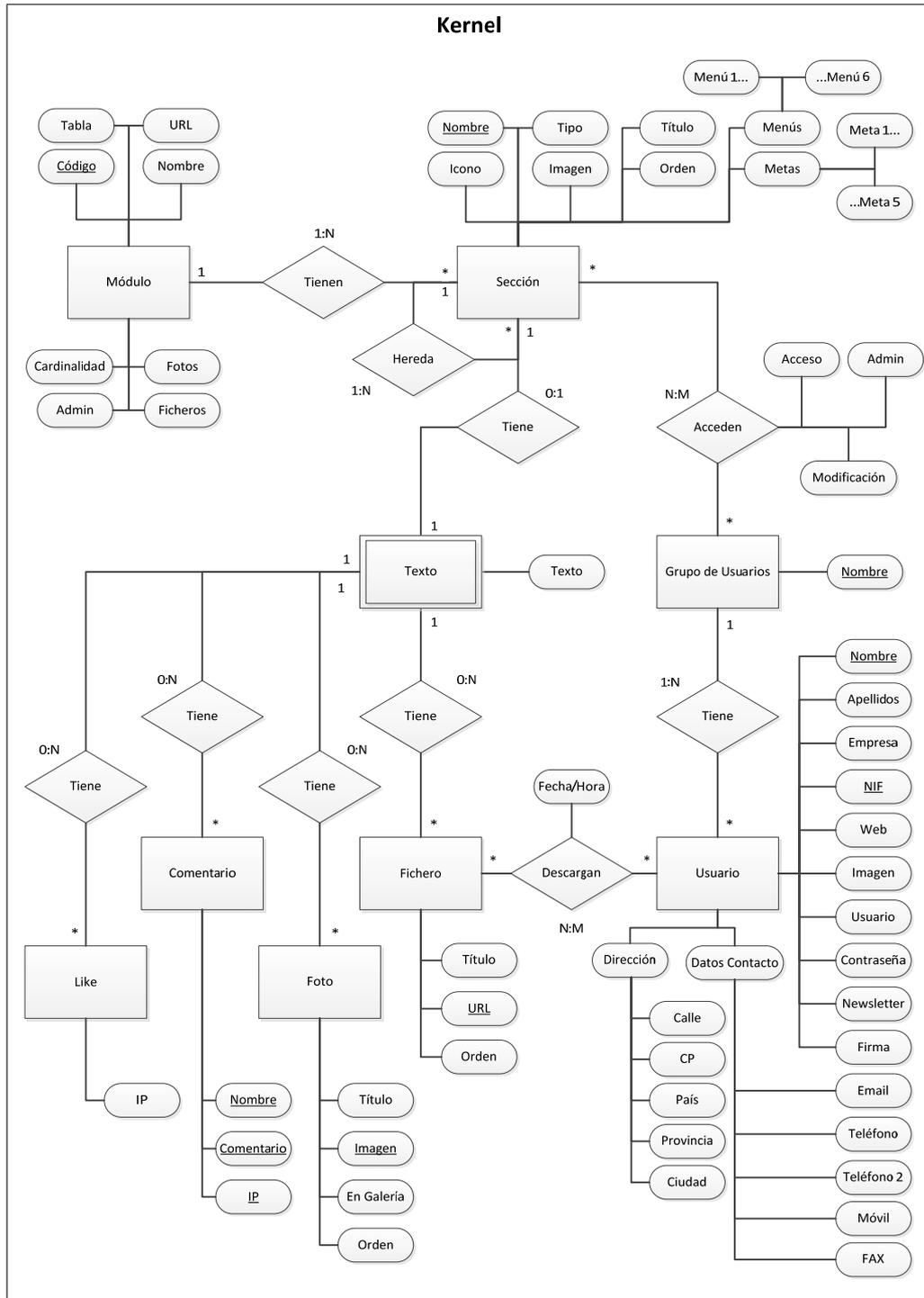


Figura -3. Modelo Entidad Relación del Kernel

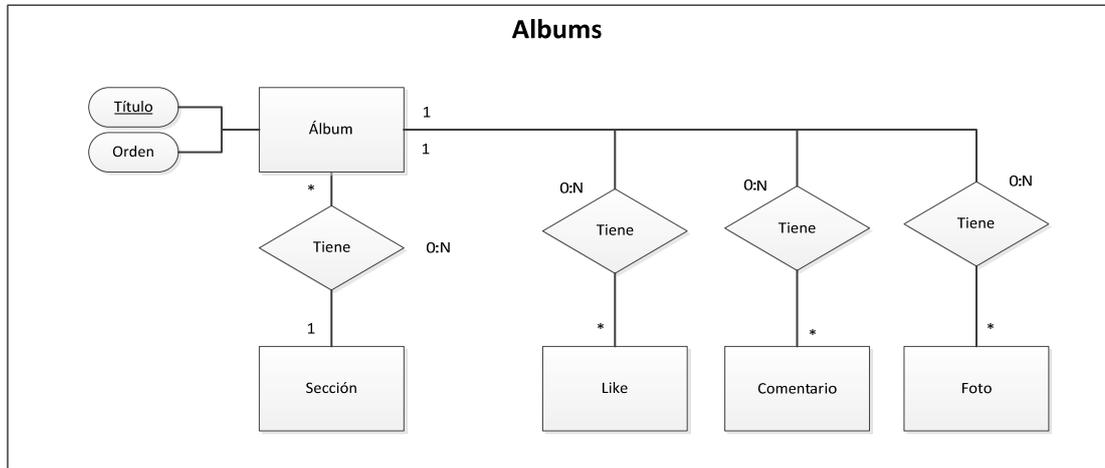


Figura -4. Modelo Entidad Relación del módulo Álbums

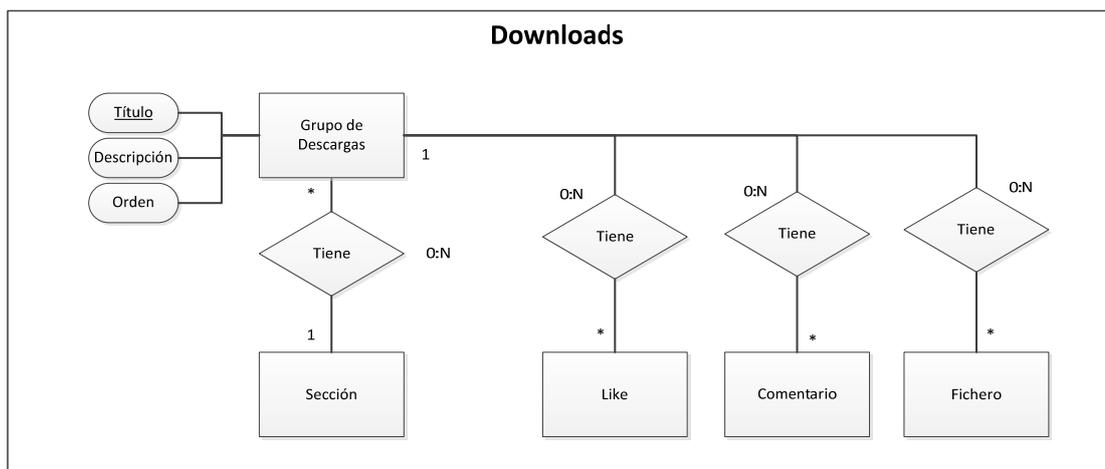


Figura -5. Modelo Entidad Relación del módulo Downloads

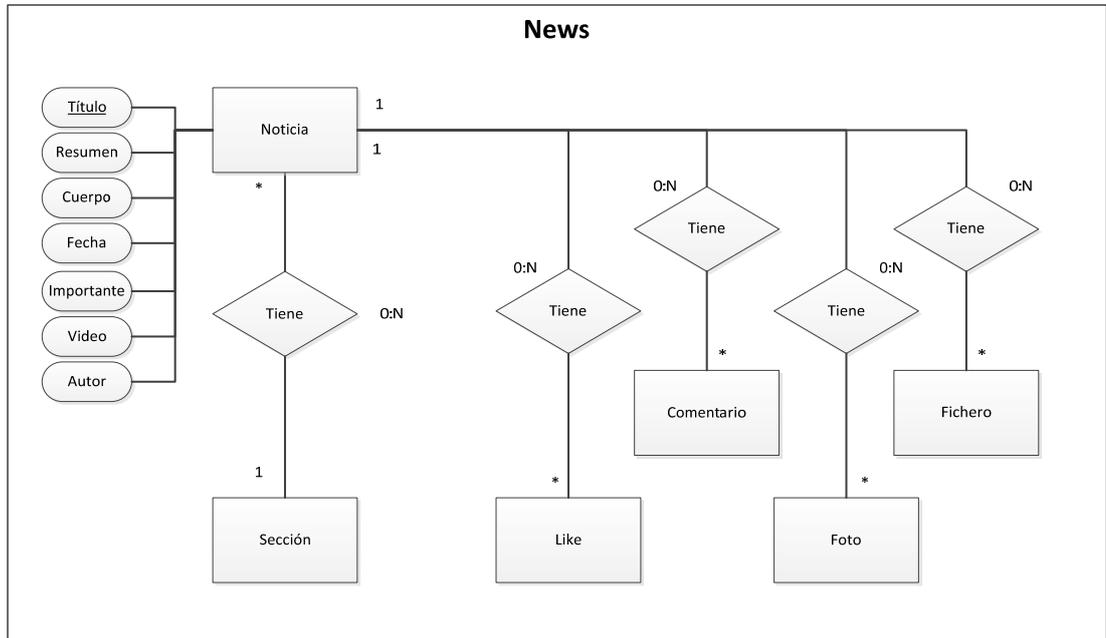


Figura -6. Modelo Entidad Relación del módulo News

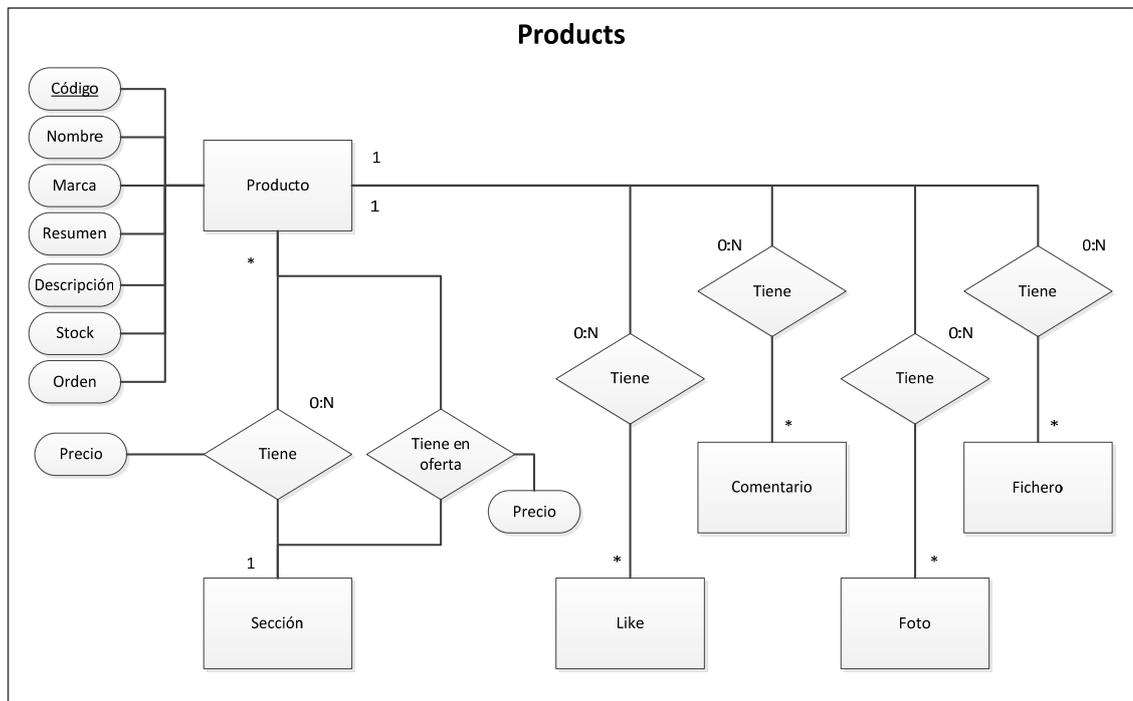


Figura 7. Modelo Entidad Relación del módulo Products

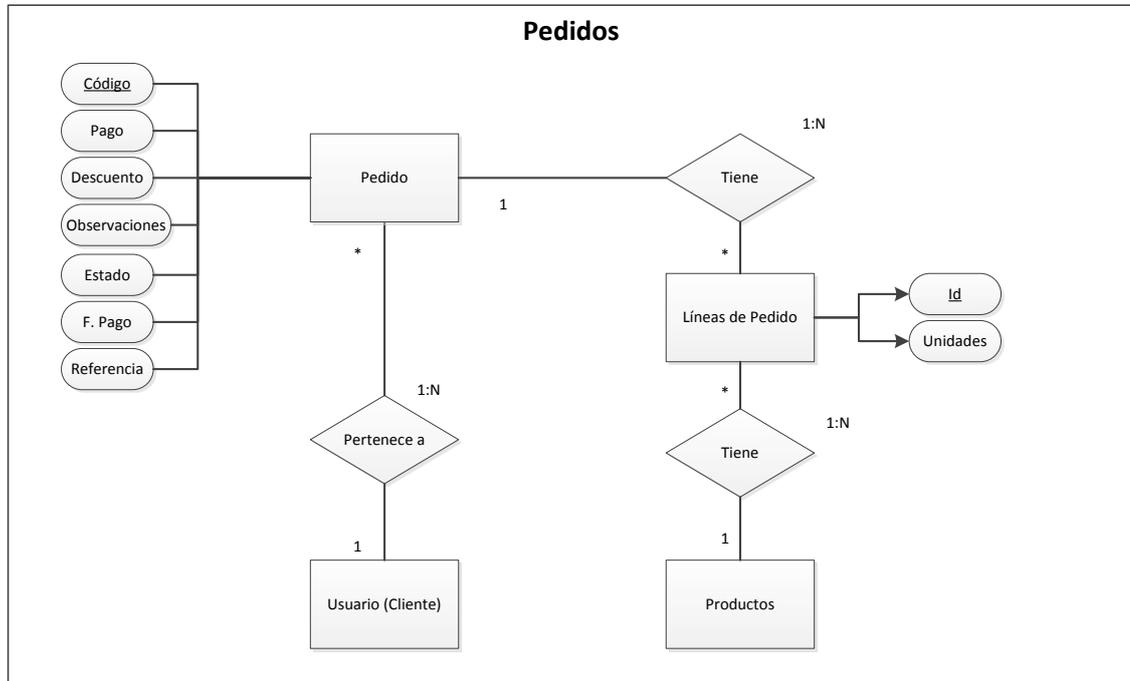


Figura 8. Modelo Entidad Relación del módulo Orders

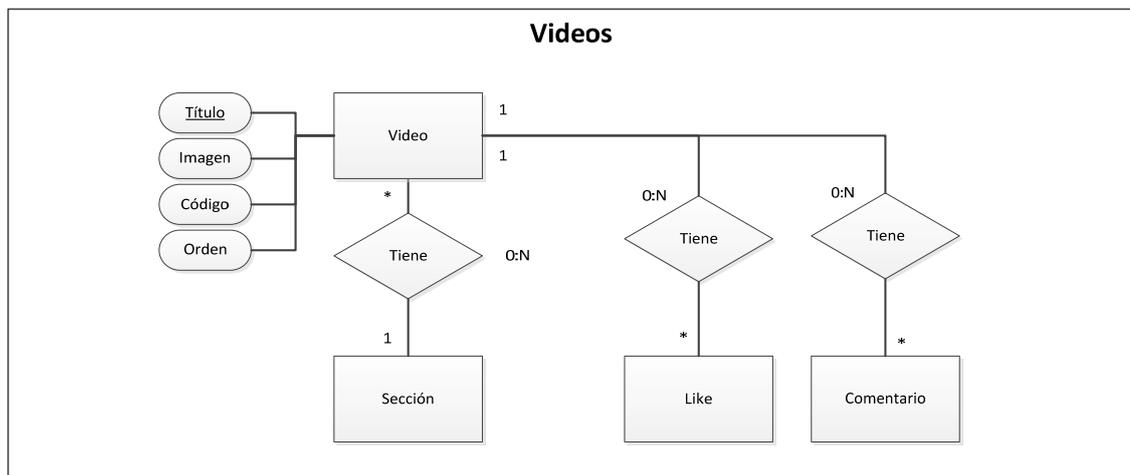


Figura 9. Modelo Entidad Relación del módulo Videos

6.2 Normalización del modelo lógico de datos

Una entidad está en 3FN si está en 2FN y todos sus atributos no principales dependen directamente de la clave primaria, es decir, no hay dependencias funcionales transitivas de atributos no principales respecto de las claves.

El modelo lógico de datos está en tercera forma normal. Todos los atributos cumplen la tercera forma normal ya que no hay dependencias entre ellos excepto con la clave primaria de su entidad. Tampoco hay dependencias funcionales transitivas de atributos no principales respecto de las claves.

7. Definición de Interfaces de Usuario

7.1 Especificación de Principios Generales de la Interface

Los principios generales de la interface HTML son los siguientes:

- *Tipo de documento:* `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- *Tipo de HTML:* `<html xmlns="http://www.w3.org/1999/xhtml">`
- *Codificación:* utf-8
- Debe cumplir los estándares W3C
- No debe contener flash
- Debe ser compatible con todos los navegadores actuales
- La navegación debe ser muy fácil e intuitiva
- No debe faltar ni sobrar opciones

El panel de administración debe estar integrado en la propia página web. Aparecerá un nuevo menú de administración en cada sección una vez introducido el usuario y contraseña.

7.2 Estructura de la Interfaces de Usuario

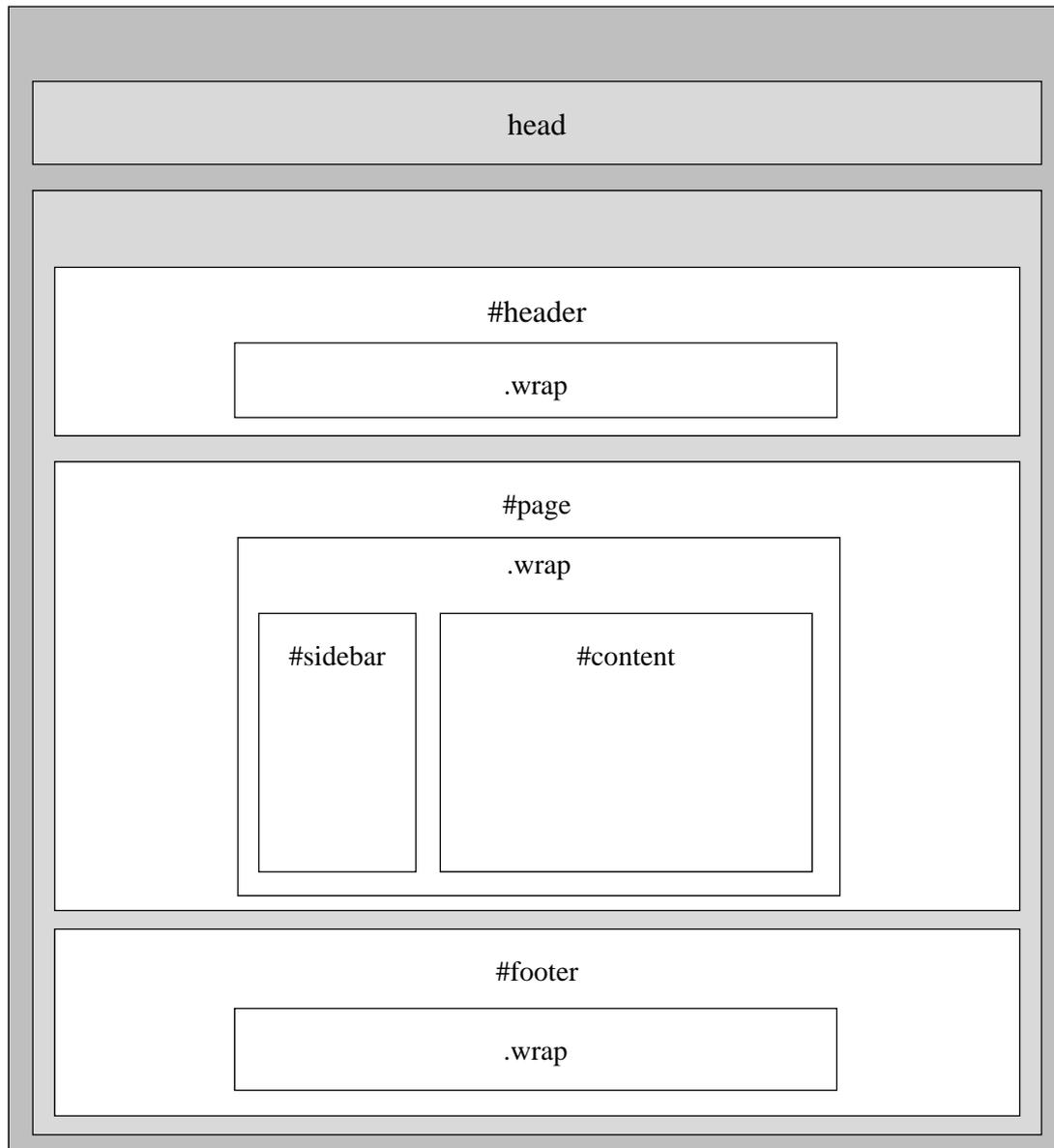


Figura 10. Estructura principal de las páginas HTML

8. Definición de la Arquitectura del Sistema

8.1 Niveles de la Arquitectura del Sistema

La arquitectura que se va a utilizar estará diseñada en tres capas. De esta forma separaremos la lógica de negocios de la lógica de diseño: Las capas serán las siguientes:

1. *Capa de acceso a datos (Data Access Layer – DAL)*: Será la encargada de acceder a los datos de la base de datos. Realizará todo el almacenamiento de datos y recibirá solicitudes de almacenamiento o recuperación de información desde la capa de negocio.
2. *Capa de Reglas del Negocio (Business Rules – BR)*: Nivel donde residirán los programas que se ejecutarán, se recibirán las peticiones del usuario y se enviarán las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecerán todas las reglas que deben cumplirse. Esta capa se comunicará con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.
3. *Capa de presentación (WEB)*: Es la capa que presentará el sistema al usuario, le mostrará la información solicitada y capturará sus datos. Esta capa se comunicará únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser *amigable* (entendible y fácil de usar) para el usuario.

Habrà una cuarta capa (*ON*) que contendrà librerías auxiliares que ayudarán a la realización de las tareas de las tres capas anteriores.

El diagrama de estas capas será el siguiente:

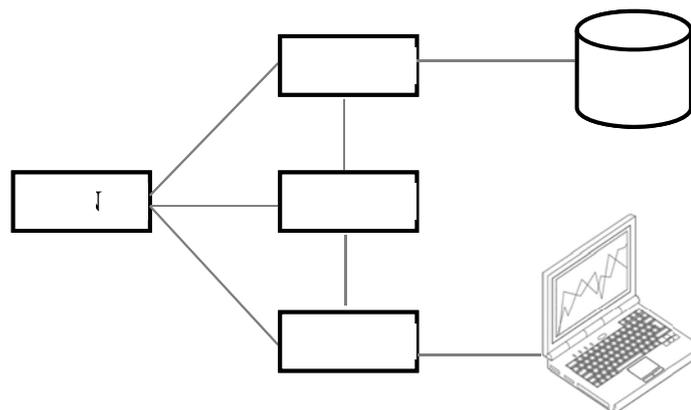


Figura 11. Diagrama de Capas

9. Lenguajes, Tecnologías y Herramientas Utilizadas

9.1 Lenguajes y Tecnologías utilizadas en el Desarrollo

- *Apache*: Servidor de páginas web HTTP, para plataformas de Windows, Unix, Linux o Macintosh, tiene la particularidad de que la arquitectura de Apache es altamente modular, por lo que se pueden añadir o quitar módulos con relativa facilidad, dependiendo de las propias necesidades del usuario (The Apache Foundation, 2014)
- *PHP*: Lenguaje interpretado en el lado del servidor. Se ha utilizado la versión 4 para el correcto funcionamiento en todos los servidores. Por este motivo, la programación no será orientada a objetos. Se ha utilizado PHP, para toda la programación en el lado del servidor, acceso a base de datos, lógica de negocio,... Se podría decir, que la verdadera 'ciencia' del proyecto ha sido realizada en este lenguaje de programación (PHP, 2015).
- *MySQL*: Sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB, desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009, desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso (MySQL, 2015).
- *HTML*: Lenguaje de marcado de texto, predominante en el desarrollo y elaboración de páginas web. Este lenguaje es el que se ha usado para definir la estructura de las páginas. El lenguaje HTML es el encargado de proporcionar un lenguaje entendible por los navegadores, como Microsoft Internet Explorer, Mozilla Firefox o Google Chrome, para que estos sean capaces de 'pintar' el contenido deseado en la pantalla. (W3SCHOOLS-HTML, 2015)
- *CSS*: Hojas de estilo en cascada. Lenguaje de programación usado para dar el formato necesario a los documentos, si con HTML definíamos la estructura de las páginas web, con las hojas de estilos se define la apariencia de las mismas, consiguiendo así una idea muy importante: separar la estructura de un documento de su presentación en pantalla (W3SCHOOLS-HTML, 2015).
- *JavaScript*: Lenguaje de programación interpretado que se ejecuta en el lado del cliente, recordad que todas las aplicaciones web están constituidas por dos partes (parte del cliente y parte del servidor). Mediante JavaScript se controla las interacciones que hace el usuario sobre la página que se encuentra cargada en ese momento en el explorador (W3SCHOOLS-HTML, 2015).
- *jQuery*: Dentro de JavaScript, cabe destacar la utilización de jQuery, framework de JavaScript, que añade funcionalidades a este lenguaje (jQuery, 2015).
- *XML*: Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un

lenguaje definido por SGML) para estructurar documentos grandes (Varios, 2014).

- **AJAX:** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML (Varios, 2012).

9.2 *Sistema Gestor de Base de Datos*

El sistema gestor de base de datos utilizado es *MySQL*. Es el más utilizado en desarrollos PHP.

MySQL es un sistema gestor de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB, desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009, desarrolla MySQL como software libre en un esquema de licenciamiento dual (Varios, 2010).

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso (Varios, 2010).

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius (Varios, 2010).

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr y YouTube (Varios, 2010).

Ventajas (Daniel, 2013):

- MySQL es Open Source
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos.
- Facilidad de configuración e instalación. Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet
- El software MySQL usa la licencia GPL

Desventajas (Daniel, 2013):

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

10. Diseño de Clases

La programación que se va a realizar no será orientada a objetos para que el gestor de contenidos funcione correctamente en todos los servidores compartidos de los principales proveedores de internet. Se programará en PHP4, pero se estructurará en ficheros como si cada uno fuese una clase.

El nombre de cada una de las clases será el nombre del módulo al que hace referencia.

Se ha diseñado un diagrama por cada capa del sistema a nivel de servidor:

1. Capa de acceso a datos (DAL)
2. Reglas del negocio (BR). Aquí se incluyen las librerías auxiliares ON

Todas las variables serán privadas, pudiendo acceder a ellas para obtener o modificar su valor, a través de funciones públicas. Así nos aseguramos que se validan los datos antes de leer o escribir en las variables.

A continuación se muestra el diagrama de clases para cada una de las capas.

10.1 Capa de Acceso a Datos (DAL)

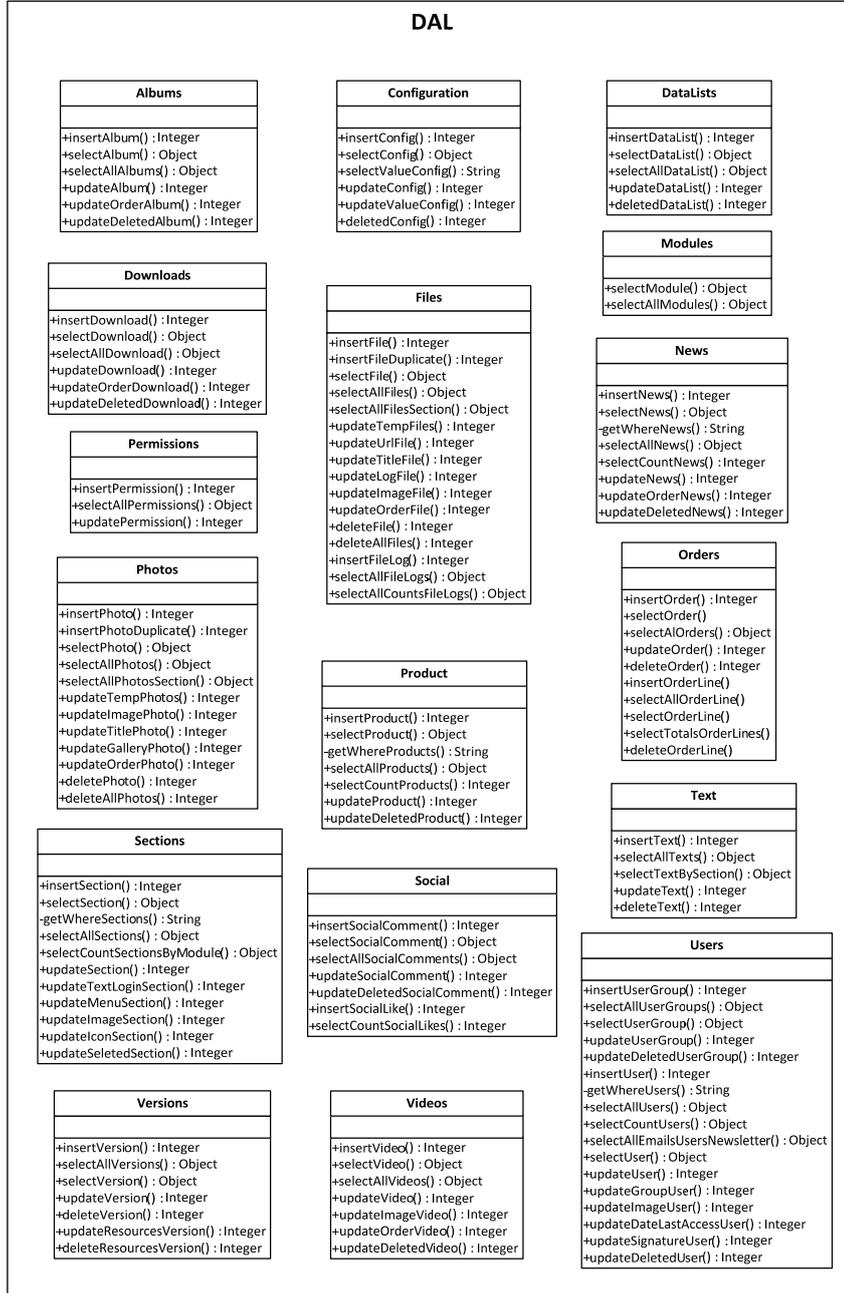


Figura 12. Diagrama de Clases de la capa DAL

10.2 Reglas del Negocio (BR)

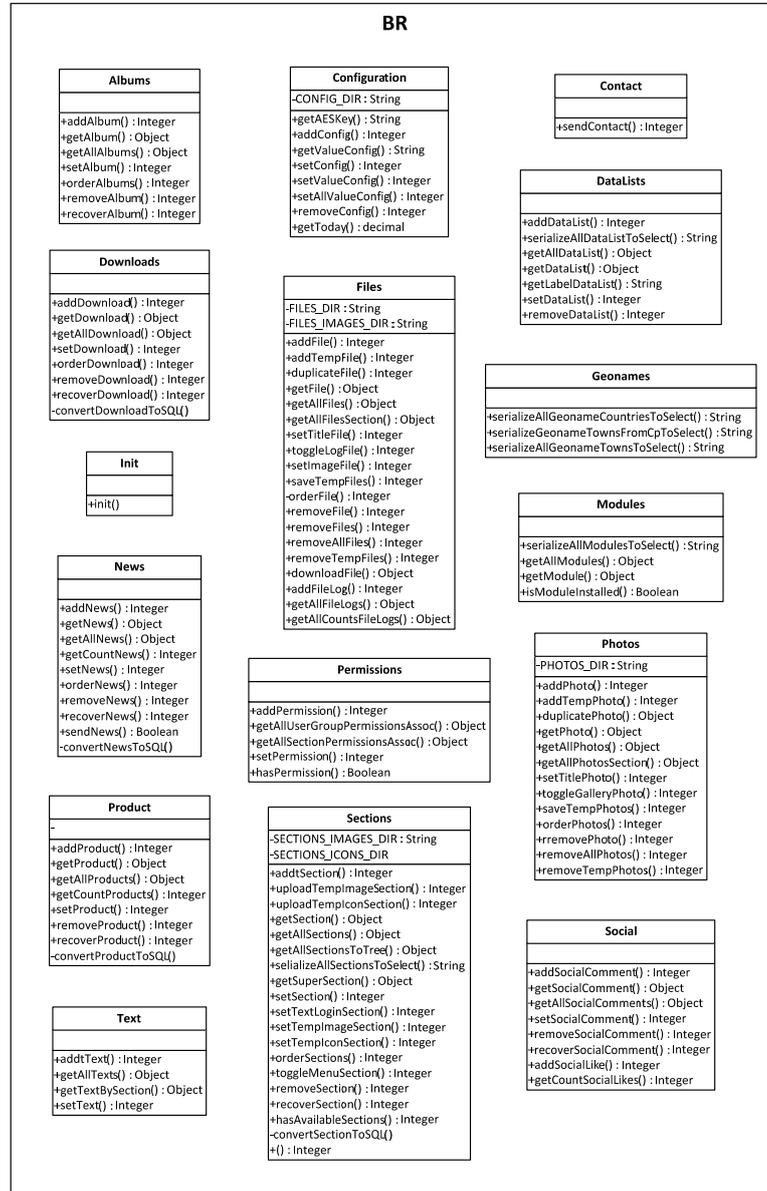


Figura 13. Diagrama de Clases de la capa BR (1/2)

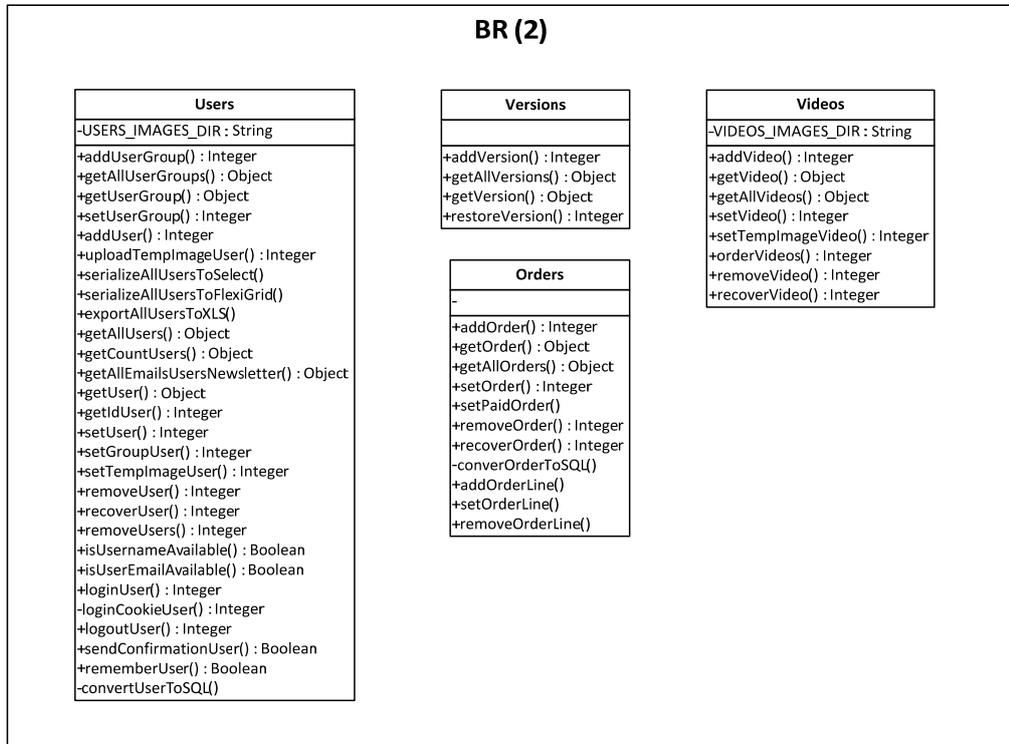


Figura 14. Diagrama de Clases de la capa BR (2/2)

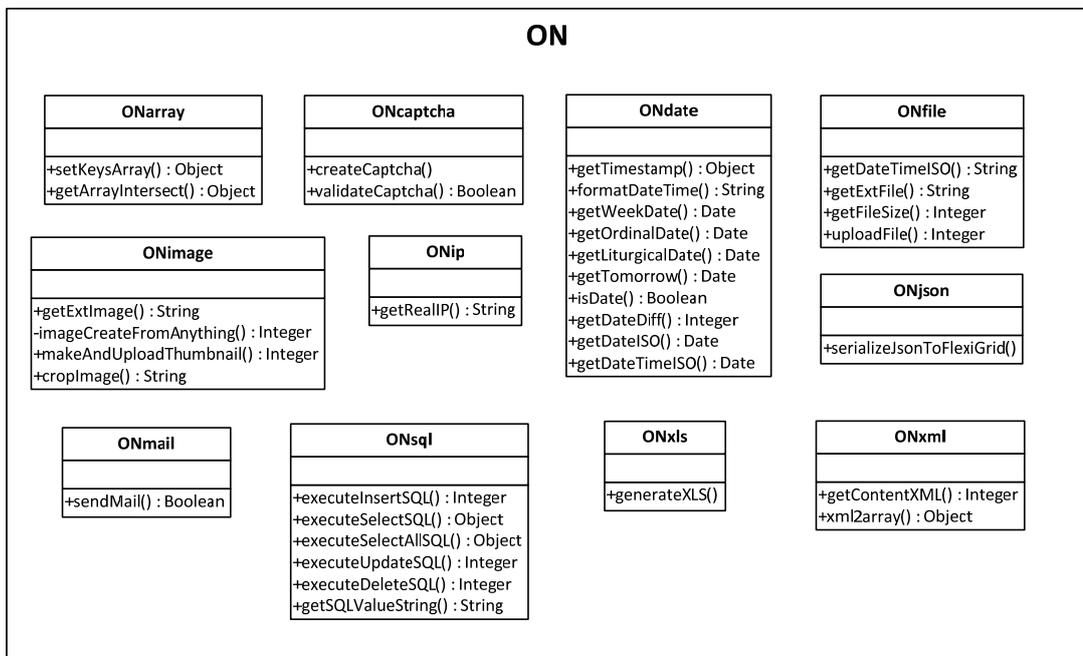


Figura 15. Diagrama de Clases de la capa ON

11. Diseño Físico de los Datos

El sistema gestor de base de datos utilizado es MySQL.

11.1 Estructura Física de la Base de Datos

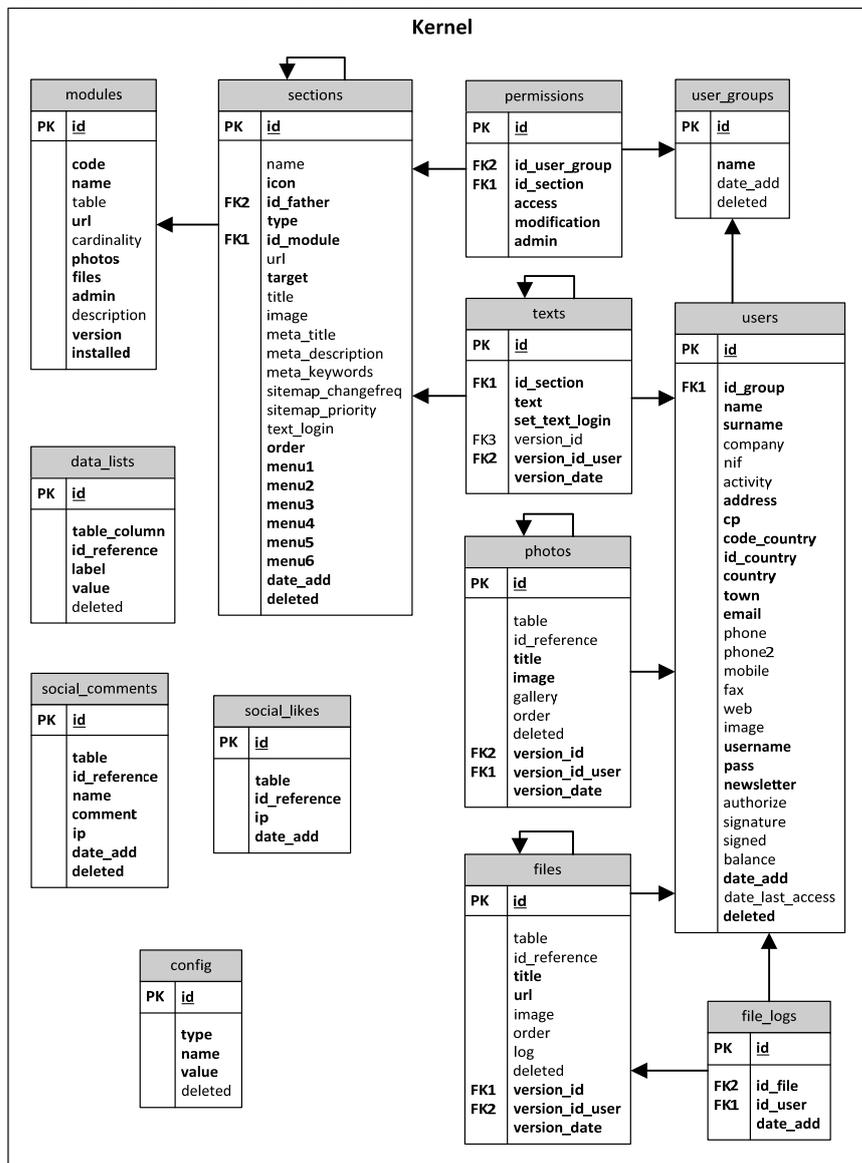


Figura 16. Estructura Física de Datos del Kernel

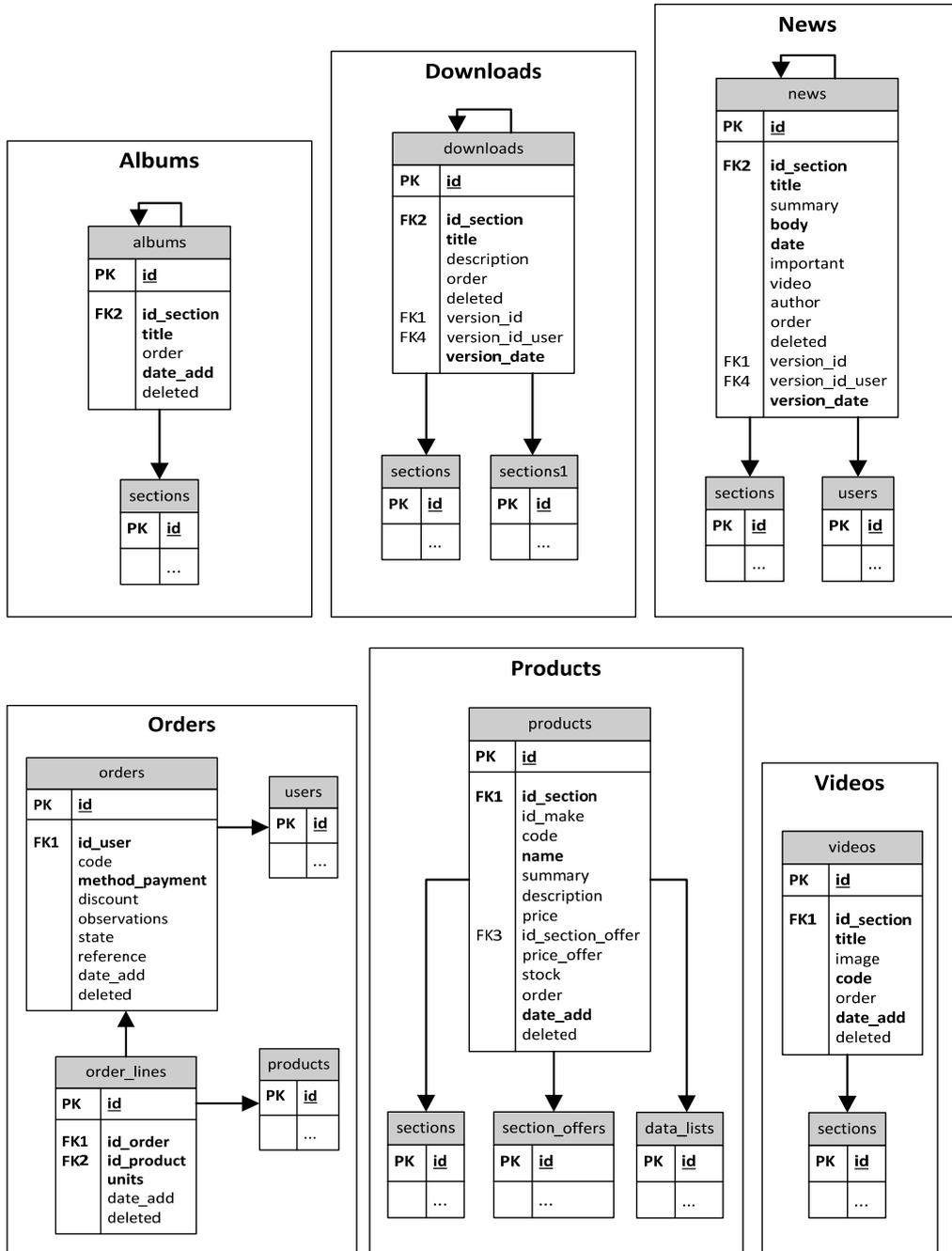


Figura 17. Estructuras Físicas de Datos de los diferentes módulos

12. Web Services para la Sincronización con ERP

Un servicio web (en inglés, Web Service o Web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet (Varios, 2012).

La sincronización de la tienda online con el ERP se realizará a través de *web services*. Así cada sistema tendrá su base de datos, y se actualizarán los datos necesarios a través de esta tecnología.

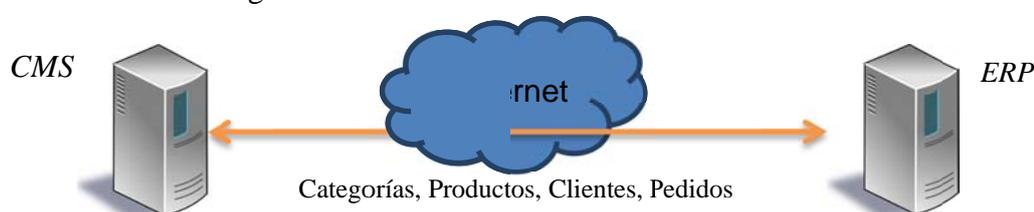


Figura 18. Diagrama de sincronización entre el CMS y el ERP

Los datos que se sincronizarán serán los siguientes:

- *Clientes*: Sincronización bidireccional, para que los clientes que se registren en la web estén automáticamente disponibles en el ERP, y viceversa.
- *Pedidos*: Sincronización bidireccional, para que los pedidos que se realicen desde la web estén disponibles en el ERP, y los pedidos que se soliciten por teléfono o en la tienda física, también puedan ser consultados desde la página web.
- *Productos*: Sincronización unidireccional, para que los productos que se añadan al ERP, estén disponibles automáticamente en la tienda online. También se controlará el inventario.
- *Categorías de productos*: Sincronización unidireccional, para que las categorías de los productos dados de alta en el ERP, estén también disponibles en la tienda online.

Para realizar la sincronización, en el lado del CMS se creará un servidor Web Services utilizando la librería de PHP NuSOAP, que está compuesto por una serie de clases que nos harán mucho más fácil la creación de este servidor.

Los métodos implementados en este servidor Web Services serán:

- *addClient*: Recibe los datos de un cliente y lo añade en la tienda online.
- *getNewClient*: Obtiene los datos de un nuevo cliente o uno modificado de la web.
- *setClient*: Recibe los datos de un cliente para modificarlo en la tienda online.
- *removeClient*: Borra un cliente en la tienda online.

- *addOrder*: Recibe los datos de un pedido y lo añade en la tienda online.
- *getNewOrder*: Obtiene los datos de un nuevo pedido o uno modificado de la web.
- *setOrder*: Recibe los datos de un cliente para modificarlo en la tienda online.
- *removeOrder*: Borra un cliente en la tienda online.
- *addOrderLine*: Recibe los datos de un detalle de pedido y lo añade en la tienda online.
- *getNewOrderLine*: Obtiene los datos de un nuevo detalle de pedido.
- *removeOrderLine*: Borra un detalle de pedido en la tienda online.
- *addProduct*: Recibe los datos de un producto y lo añade en la tienda online.
- *setProduct*: Recibe los datos de un producto para modificarlo en la tienda online.
- *removeProduct*: Borra un producto en la tienda online.
- *addSection*: Recibe los datos de una sección y lo añade en la tienda online.
- *setSection*: Recibe los datos de una sección para modificarlo en la tienda online.
- *removeSection*: Borra un detalle de pedido en la tienda online.

En el ERP, se creará un *plugin* con el cliente de *Web Services*, que únicamente llamará a los servicios web del CMS, para obtener o enviar datos desde y hasta la tienda online.

13. Establecimiento del Plan de Implantación

Para implantar el CMS en un entorno real no hace falta que sea un servidor dedicado, bastaría con contratar un *hosting* con *PHP* y *mysql*. El espacio en disco mínimo recomendado es de 500 MB, aunque si la página web va a tener muchas imágenes y ficheros de descarga, el espacio debería ser mayor.

También hace falta un dominio asociado a ese alojamiento para poder acceder a la página web desde cualquier ordenador conectado a internet.

Cuando ya esté el *hosting* preparado, bastará con subir todos los archivos por FTP a la carpeta pública a la que apunte el dominio, e importar la estructura y datos a la base de datos del servidor.

Por último, habría que editar la conexión de la base de datos para que coincida con la creada en el servidor real.

14. Conclusiones y Trabajo Futuro

La elección de desarrollar un gestor de contenidos propio con todas las alternativas de software libre que hay en el mercado no es una decisión fácil, pero después de estudiar las alternativas más conocidas, ninguna se ajustaba a todas las necesidades.

Todas las soluciones estudiadas destacan en algún aspecto, pero fallaban en otros, por lo que elegir cualquier *CMS* del mercado, en algún momento se va a quedar corto para alguna característica.

Después de más de treinta desarrollos web utilizando este gestor de contenidos, no ha habido ningún caso en el que no se pudiera implementar alguna característica que quisiera o necesitara.

Además, gracias a un gestor de contenidos propio se consigue diferenciarse del mercado, en el cual hay muchísimas empresas que dicen desarrollar páginas web y que lo único que hacen es descargarse un gestor de contenidos, una plantilla y poner los textos y logotipos; y si les piden algo en concreto, como no haya un *plugin* ya desarrollado para ese *CMS*, ni siquiera sabrían desarrollar uno ellos.

Con un gestor de contenidos en el que sepa al cien por cien cómo está programado, nunca se va a tener el problema de no saber dónde tocar para añadir una nueva funcionalidad, o el miedo de que con una nueva actualización del sistema se pierdan las modificaciones que se hayan realizado. Además, como el código fuente no estará disponible en internet, se consigue un sistema mucho más seguro que los demás. Y como está desarrollado a medida, ni sobran ni faltan líneas de código ni funcionalidades, por lo que las cargas son mucho más rápidas y su funcionamiento para es mucho más sencillo.

Pero lo más importante es lo gratificante que resulta realizar páginas web habiéndolas desarrollado desde cero, y conseguir te feliciten porque se consigue una página web intuitiva y que se ajuste a las necesidades de cada desarrollo.

Como trabajo actual y futuro, destacaría los siguientes puntos:

- Seguir mejorando y añadiendo módulos al gestor de contenidos
- Programarlo en *HTML5* y *CSS3*.
- Diseñarlo utilizando *Responsive Design* para que se adapte a todos los tamaños de pantalla de ordenadores, *tablets* y *smartphones*.

15. Bibliografía

- CONSULTRANS (2014): “Estudio sobre el uso e impacto de los gestores de contenidos en la modernización e innovación de las PYME españolas”, en Madrimasd, accesible en http://www.madrimasd.org/informacionIDI/noticias/documentos/Presentacion_Resultados_Estudio_Gestores_Contentidos.pdf
- DANIEL (2013): “Ventajas Y Desventajas” en MySQLDaniel [en línea], accesible en <http://mysqldaniel.wordpress.com/ventajas-y-desventajas/>
- GARCÍA CUERVA, Xavier (2014): “Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto”, en Mosaix [en línea], accesible en <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>
- jQuery (-2015): “jQuery Documentattion” en jQuery [en línea], accesible en <http://jquery.com/>
- MySQL (-2015): “MySQL Documentation” en MySQL [en línea], accesible en <http://www.mysql.com/>
- PHP (-2015): “PHP Documentation” en PHP [en línea], accesible en <http://php.net/>
- THE APACHE SOFTWARE FOUNDATION (-2015): “Apache Documentation” en Apache [en línea], accesible en <http://www.apache.org/>
- VARIOS (2010-2015): “MySQL “ en Wikipedia [en línea], accesible en <http://es.wikipedia.org/wiki/MySQL>
- VARIOS (2012-2015): “AJAX “ en Wikipedia [en línea], accesible en <http://es.wikipedia.org/wiki/AJAX>
- VARIOS (2012-2015): “Servicio Web” en Wikipedia [en línea], accesible en http://es.wikipedia.org/wiki/Servicio_web
- VARIOS (2013-2015): “Sistema de gestión de contenidos” , en Wikipedia [en línea], accesible en http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos
- VARIOS (2014-2015): “Extensible Markup Language “ en Wikipedia [en línea], accesible en http://es.wikipedia.org/wiki/Extensible_Markup_Language
- W3SCHOOLS-HTML (-2015): “HTML Tutorial” en W3Schools [en línea], accesible en <http://www.w3schools.com/html/>
- W3SCHOOLS-CSS (-2015): “CSS Tutorial” en W3Schools [en línea], accesible en <http://www.w3schools.com/css/>
- W3SCHOOLS (-2015): “JS Tutorial” en W3Schools [en línea], accesible en <http://www.w3schools.com/js/>