

Universidad de las Ciencias Informáticas
Facultad 1



*Módulo para la administración de contenidos en
aplicaciones web desarrolladas con Drupal 7*

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor:

Anabel García García

Tutores:

Ing. Yaima Oval Riverón
Ing. Ramón Morales Alvarez

Ciudad de La Habana, Cuba

Junio, 2017



Pensamiento

"Crear que el trabajo constante, firme, e infatigable puede obtenerlo todo, constituye la imposibilidad del fracaso".

Henry Ford

Declaración de autoría

Declaro por este medio que yo Anabel García García, con carné de identidad 93072007695 soy el autor principal del trabajo titulado “Módulo para la administración de contenidos en aplicaciones web desarrolladas con Drupal 7” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del 2017.

Autor: Anabel García García

Tutor: Ing. Yaima Oval Riverón

Tutor: Ing. Ramón Morales Alvarez

Dedicatoria

Dedicado a mi madre, a mi tío y a mi padrastro por ser mi apoyo en estos 5 años de universidad y darme todo lo que necesité.

Agradecimientos

Agradezco a toda mi familia, en especial a mi madre y mi padrastro que siempre han estado ahí para mí, igual que mi tío que aunque no esté cerca he sentido su apoyo todo este tiempo. También quiero agradecer a mis amigos, Alejandra, Claudia y Ricardo por darme ánimos cuando más lo necesité y a todos los amigos que conocí en la escuela en especial con los que conviví estos últimos años.

Resumen

El objetivo principal de esta investigación es desarrollar un módulo para Drupal que perfeccione la administración de contenidos de los sitios web. Surge ante la necesidad de mejorar dicha administración, hacer más ágil la valoración de las estadísticas para posible análisis de parte de los administradores y realizar notificaciones a los usuarios mediante correo electrónicos. Actualmente el CMS Drupal no cuenta con un módulo que cumpla estas funcionalidades, lo que trae consigo que los usuarios y administradores no puedan realizar de manera más eficaz la administración de contenidos. Para el desarrollo de la propuesta de solución se utiliza el CMS Drupal 7, Netbeans, Visual Paradigm. Se emplea la metodología de desarrollo AUP en su versión UCI para guiar todo el proceso de desarrollo de la aplicación. A partir de la realización de este trabajo se obtuvo como resultado un módulo que resuelve la problemática antes planteada.

Palabras clave: administración, contenido, Drupal, módulo.

Índice

Índice	7
Introducción	9
Capítulo 1. La gestión de contenidos en aplicaciones web desarrolladas con Drupal 7.	13
1.1 Historia de los CMS	13
1.1.1 Clasificación de los CMS	14
1.2 Necesidad de un CMS	15
1.3 Sistemas de Gestión de Contenidos	16
1.4 Drupal	18
1.5 Funcionalidades de Drupal	18
1.6 Modularización del crecimiento	19
1.7 Módulos de Drupal 7	21
1.8 Herramientas utilizadas para al desarrollo	24
1.9 Lenguajes de programación	27
1.10 Tecnologías utilizadas para al desarrollo	29
1.11 Metodología de desarrollo.....	29
1.12 Conclusiones.....	31
Capítulo 2. Módulo para el CMS Drupal 7.	32
2.1 Descripción de la propuesta de solución	32
2.2 Modelo de dominio	32
2.3 Roles y responsabilidades	33
2.4 Requisitos funcionales.....	34
2.5 Requisitos no funcionales	35
2.6 Historias de usuario	36
2.7 Modelo de datos	40
2.8 Arquitectura de Drupal.....	40
2.9 Patrones de diseño	42
2.10 Modelo de despliegue.....	44

2.11	Diagrama de componentes.....	45
2.14	Conclusiones.....	46
Capítulo 3. Validación del Módulo para el CMS Drupal.		47
3.1	Estándares de codificación	47
3.2	Pantallas del módulo de administración de contenidos en aplicaciones web desarrolladas con Drupal	53
3.3	Validación de la propuesta de solución	54
3.4	Conclusiones.....	61
Conclusiones Generales.....		62
Recomendaciones		63
Bibliografía		64
Glosario de términos		67
Anexos.....		70

Introducción

La sociedad se encuentra en constante cambio y desarrollo proceso estrechamente vinculado al surgimiento continuo de nuevas formas de producción, relaciones sociales y tecnologías. De manera que, las Tecnologías de la Información y las Comunicaciones (TIC) se han convertido a una gran velocidad, en parte importante de la humanidad. Este proceso se debe principalmente al surgimiento de la Internet.

El desarrollo de Internet está basado en las prestaciones de servicios de uso común para los usuarios. Por las facilidades que brinda la web, su uso se ha multiplicado aceleradamente en los últimos años, lo que implica que el desarrollo de este tipo de aplicaciones se haya elevado exponencialmente (Hume, 2015).

Para la realización de un servicio o sistema Web los desarrolladores necesitan tiempo suficiente para diseñar y programar la implementación de estos servicios o sistemas. Con la propia evolución de las tecnologías han surgido herramientas que permiten establecer una estructura de soporte para la creación y administración de contenidos por parte de los desarrolladores, editores y demás usuarios de los sitios web. Estas herramientas se denominan Sistema de Gestión de Contenido (Content Management System o CMS) y tienen dentro de sus funcionalidades el manejo del contenido, además la posibilidad de cambiar el diseño del sitio web sin tener que darle formato al contenido de nuevo y permitir a varios editores de forma sencilla y controlada, la publicación en el sitio (Cuerda, 2004).

Vinculada al proceso de informatización y desarrollo de la industria del software, la Universidad de las Ciencias Informáticas (UCI), tiene un programa de estudio donde se vincula la práctica profesional con la docencia y cuenta con la tecnología necesaria para lograr la formación de profesionales. Uno de los centros de desarrollo perteneciente a la UCI es el Centro de Ideoinformática (CIDI) que pertenece a la facultad 1. Este centro tiene dentro de sus misiones, el desarrollo web y utiliza el CMS Drupal para la gestión de los contenidos de estos portales: responsabilidad asociada al departamento de Servicios Informáticos para Internet (SENIT).

En SENIT los desarrolladores pueden facilitar su trabajo filtrando los contenidos por algunas de sus características, que son: estado, tipo de idioma, y estados opuestos a la vez (estos pueden ser publicados y no publicados), pero están limitados ya que esta sección no permite filtrar por título, rango de fecha de creación y edición y tampoco por usuarios.

Al realizar la búsqueda de contenidos, a los administradores, se les muestra en una lista todos los contenidos encontrados. Cuando la cantidad es significativa se les hace un poco engorrosa la tarea de encontrar el contenido deseado ya que Drupal no permite establecer la cantidad de elementos que se van a mostrar por página al realizar una búsqueda. Drupal no muestra gráficas estadísticas sobre los contenidos creados, tampoco permite personalizar la lista de contenidos y no realiza notificaciones mediante correo electrónico a los usuarios cuando se crea, modifica o elimina algún contenido de su autoría.

Lo antes planteado contribuyen a que el proceso de administración de los contenidos por parte de los administradores de los portales web desarrollados con Drupal, sea lento y engoroso a la hora de realizar búsquedas de contenidos específicos para su edición; así como para garantizar el seguimiento a los contenidos creados porque no hay notificaciones automática de los cambios que se vayan realizando en los mismos.

Además, la carencia de las estadísticas relacionadas con la actividad de estos portales en cuanto a la administración de los contenidos dificulta la toma de decisiones acertadas en función de los tipos de contenidos de mayor y menor creación por parte de los administradores de los portales web desarrollados con Drupal, y valorar el comportamiento de estos para posibles análisis.

Para darle solución a la situación planteada se concibe como **problema científico** ¿Cómo perfeccionar la administración de contenidos en aplicaciones web desarrolladas con Drupal 7?

Por lo tanto, el **objeto de estudio** de esta investigación es el proceso de la administración de contenidos en aplicaciones web.

El **objetivo general** de este trabajo es desarrollar un módulo para perfeccionar la administración de contenidos en aplicaciones web desarrolladas con Drupal 7.

De ello se deriva que el **campo de acción** que abarca este trabajo es la administración de contenidos en aplicaciones web desarrolladas con Drupal 7.

Para dar cumplimiento al objetivo se plantean las siguientes preguntas científicas:

Preguntas científicas

¿Cuáles son los supuestos teóricos que sustentan la implementación de un módulo para la administración de contenidos?

¿Qué elementos presenta el análisis y diseño del módulo que se encarga de la administración de contenidos en Drupal 7?

¿Qué concepción debe tener el módulo que se encarga de la administración de contenidos en Drupal 7?

¿Cómo validar el módulo para la administración de contenidos en Drupal 7?

Para dar respuestas a las preguntas anteriormente planteadas se definen las siguientes tareas:

1. Sistematización teórica sobre la administración de contenidos en aplicaciones web utilizando CMS.
2. Definición de las funcionalidades del módulo para la administración de contenidos en Drupal 7.
3. Elaboración del diseño de los diagramas necesarios para dar solución a la problemática planteada.
4. Implementación de las funcionalidades del módulo de gestión de contenidos en aplicaciones web.
5. Realización de pruebas para la verificación del funcionamiento del módulo de gestión de contenidos en aplicaciones web desarrolladas con Drupal 7.

Para realizar las tareas de investigación se emplean los siguientes métodos científicos de la investigación:

Métodos teóricos:

Análisis Histórico-Lógico: este método permitió estudiar el avance y las novedades que ha tenido el proceso de desarrollo de módulos y perfiles de instalación a través de las diferentes versiones del CMS Drupal y su aplicación en distintos sistemas web.

Análítico-Sintético: este método permitió seleccionar los elementos más importantes relacionados con módulos y perfiles de instalación, necesarios para el desarrollo de portales web utilizando Drupal.

Métodos empíricos:

Observación: este método ayudó a definir características como filtrar contenido, establecer la cantidad de elementos que se van a mostrar al realizar la búsqueda, mostrar contenidos propios, mostrar estadísticas sobre los contenidos creados y retroalimentación automática de los cambios que se vayan realizando en los mismos, entre otros.

Modelación: Permitted modelar los diagramas necesarios para obtener una idea más acertada acerca del análisis, diseño e implementación del módulo.

La presente investigación está estructurada en tres capítulos:

Capítulo 1: La administración de contenidos en aplicaciones web desarrolladas con Drupal 7.

Se muestran las características de los CMS de manera general e información de los más utilizados en la actualidad. Además se profundiza en las particularidades de Drupal, principalmente las relacionadas con los lenguajes y herramientas con los que se encuentra vinculado el mismo.

Capítulo 2: Módulo para el CMS Drupal 7.

En este capítulo se realiza la modelación de la solución propuesta, además se identifican los requisitos funcionales y no funcionales, así como las historias de usuario correspondiente a los mismos. También se muestran los artefactos de la metodología de desarrollo de software seleccionada, se describen los diagramas correspondientes al diseño del módulo y el modelo de datos.

Capítulo 3: Validación del Módulo para el CMS Drupal 7

En este capítulo se abordan aspectos relacionados con la implementación del sistema en base a la arquitectura de desarrollo de software definida. Se documentan las pruebas de software realizadas al módulo de gestión de contenidos en aplicaciones web desarrolladas con Drupal.

Capítulo 1. La gestión de contenidos en aplicaciones web desarrolladas con Drupal 7.

En el presente capítulo se brinda una visión general de los aspectos relacionados con los Sistemas de Administración de Contenido (CMS), las características de cada herramienta necesaria para el desarrollo del módulo, así como los principales conceptos asociados al dominio del problema y que son necesarios para entender la propuesta de solución.

1.1 Historia de los CMS

A principios de los años noventa, el concepto de sistemas de gestión de contenidos era desconocido. Algunas de sus funciones se realizaban con aplicaciones independientes: editores de texto y de imágenes, bases de datos y programación a la medida. En año 1994 *Illustra Information Technology*¹ utilizaba una base de datos de objetos como repositorio de los contenidos de una web, con el objetivo de poder reutilizar los objetos y ofrecía a los autores un entorno para la creación basado en patrones. La idea no tuvo aceptación por el público y por tanto no se continuó con su implementación. La parte de la empresa enfocada a la web fue comprada por AOL², mientras que Informix³ adquirió la parte de bases de datos. En el año 1994 RedDot⁴, la cual es una de las empresas pioneras, empezó el desarrollo de un gestor de contenidos. No fue hasta a finales del año siguiente que presentaron su CMS basado en una base de datos. Entre los primeros CMS de código abierto se encuentra Typo 3, que empezó su desarrollo el año 1997 por su autor, Kasper Skarhoj. PHPNuke, la herramienta que popularizó el uso de estos sistemas para las

1 *Illustra Information Technologies*, es un desarrollador de software de gestión de base de datos relacional.

2 Anteriormente conocida como *America Online*, es una empresa de servicios de internet y medios con sede en Nueva York. Ha franquiciado sus servicios a empresas en varios países alrededor del mundo o establecido versiones internacionales de sus servicios.

3 Es una familia de productos de sistema de gestión de bases de datos relacionales de IBM, adquirida en 2001 por una compañía (también llamada Informix o Informix Software) cuyos orígenes se remontan a 1980.

4 RedDot, fundada en 1993, es una unidad de negocios de *Open Text Corporation* y se conoce como el Grupo de Soluciones Web de *Open Text*. El software ayuda en la gestión de contenidos, con cumplimiento normativo y requisitos específicos de la industria.

comunidades de usuarios en Internet, se empezó a desarrollar en año 2000. La primera versión supuso tres semanas de trabajo al creador, reescribiendo el código de otra herramienta, Thatware (Cuerda,2004).

El trabajo relacionado a la web puede ser complicado y muy laborioso si no se conoce del uso de herramientas adecuadas. En los últimos años se ha creado y desarrollado el tema de sistema de gestión de contenidos dando paso al surgimiento de herramientas basadas en esto, un ejemplo son los CMS (*Content Management Systems*) (Cuerda, 2004).

Los Sistemas Gestores de Contenidos son aplicaciones web especialmente diseñadas para permitir crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web. Las aplicaciones web son aquellas aplicaciones a las que se accede a través de un navegador web. Los gestores de contenidos o CMS como aplicaciones web que son, habitualmente necesitan de la compañía de una serie de elementos que facilitan su funcionamiento:

- 1) Un servidor web: encargado de recibir las peticiones de los navegadores web de los clientes cuando solicitan una página web, de comunicarse con el módulo encargado de la ejecución del código y de enviar las páginas web resultado de la ejecución del código al navegador del cliente.
- 2) Módulo: encargado de ejecutar el código escrito en un lenguaje de programación y de enviar la página web resultante al servidor web (para la mayoría de CMS se utiliza el módulo PHP del servidor Apache)
- 3) Un servidor de base de datos: encargado de almacenar los datos del sitio web. Uno de los más utilizados en los gestores de contenidos es el servidor de base de datos MySQL.
- 4) Un lenguaje de programación: el lenguaje de programación más utilizado para los gestores de contenido más populares es PHP (Barberá, 2014).

1.1.1 Clasificación de los CMS

Los CMS pueden ser clasificados atendiendo a sus características:

1) Según el lenguaje de programación empleado por el CMS para crear la página web, por ejemplo: Java, PHP, ASP.NET, Python, PERL. Los más conocidos gestores de contenidos como Drupal, Wordpress, entre otros están codificados en el lenguaje de programación del lado del servidor PHP.

2) Según la licencia: Existen CMS cuyo código se puede leer, modificar y redistribuir gratuitamente llamados CMS de código abierto, mientras los que no lo permiten se conocen como CMS de software propietario.

También pueden ser clasificados por su uso y funcionalidad como:

1) Genéricos: Los CMS pueden usarse para crear una página web corporativa, un blog o una tienda online. Aquí se puede incluir los CMS: Joomla! y Drupal.

2) Blogs: son los CMS especialmente creados para la gestión de diarios personales. Ejemplo WordPress que comenzó siendo un gestor de contenidos específico para la creación de Blogs.

3) Comercio electrónico: son CMS creados específicamente para crear tiendas online. Algunos ejemplos son: Magento, PrestaShop y Opencart.

4) Existen CMS específicos para crear Foros, Wikis o CMS para cursos online Moodle (Barberá, 2014).

1.2 Necesidad de un CMS

El desarrollo basado en un CMS tiene una gran cantidad de ventajas que hacen que sea una opción muy recomendable en el momento de plantearse la creación de una aplicación web:

- Los CMS tienen funcionalidades concretas sin necesidad de arreglar fallos en cuestiones básicas y permitiendo centrarse en aspectos más complejos.
- Suelen cubrir necesidades para plataformas de comercio electrónico, blogs, páginas con contenido estructurado y gestión de usuarios.
- Un usuario puede gestionar su propia web después de recibir formación. Una vez desarrollada la web, el propio usuario puede añadir, editar o borrar datos y mantener su página viva por sí mismo.
- El desarrollo es más rápido y limpio. La creación de plantillas y funcionalidades para los gestores de contenidos se basan en unos estándares que un buen equipo es capaz de seguir para conseguir los resultados deseados partiendo de una base sólida ya constituida. Esto elimina en gran parte el

esfuerzo necesario para construir el esqueleto de código de una web.

- Permiten desarrollos escalables. Esto significa que, con una buena planificación, se puede orientar el desarrollo desde un CMS para añadir nuevas funcionalidades no previstas en el futuro sin que la web sufra cambios traumáticos mediante la incorporación de módulos.
- Muchos desarrolladores utilizan los CMS más populares. Esto es una ventaja importante, porque cualquier programador es capaz de hacerse cargo y modificar cualquier web siempre que conozca la arquitectura del CMS empleado aunque no sean ellos los que la hayan desarrollado.
- Muchos creadores de contenido saben manejarlos. Es relativamente fácil encontrar personas que pueden hacerse cargo del contenido de una web desarrollada a partir de un CMS sin necesidad de enseñarles cómo se maneja.
- Cada CMS tiene una comunidad de programadores detrás que constantemente soluciona fallos, crea nuevos módulos y están en contacto unos con otros para proporcionar el mejor servicio posible a los clientes para los que desarrollan (Trazada, 2013).

1.3 Sistemas de Gestión de Contenidos

Entre las soluciones de código abierto más populares y conocidas se encuentran: WordPress, Joomla!, TYPO3, Contao y Drupal. Con cualquiera de estos sistemas es posible ejecutar proyectos de cualquier tipo. Esta lista podría ser complementada fácilmente por otros gestores de contenido; sin embargo, este estudio estará centrado en los mencionados anteriormente, pues su cuota total de uso en el mercado está alrededor del 85 por ciento (1and1, 2016).

Joomla!

Joomla! es uno de los más populares paquetes de software usado para crear, organizar, administrar y publicar contenido para sitios web, blogs, intranets y aplicaciones móviles. Debido a su arquitectura Modelo-Vista-Controlador (MVC) escalable es también una base ideal para crear aplicaciones web.

Según el sitio oficial de Joomla!, más del 3 por ciento de la web se ejecuta con este CMS, el cual también potencia la presencia en Internet de cientos de miles de pequeñas empresas, gobiernos, organizaciones sin fines de lucro y grandes organizaciones de todo el mundo. Este CMS dirigido por una comunidad internacional de más de un millón y medio de contribuyentes activos, ayuda a que desde el usuario más

inexperto de la web al más experimentado desarrollador, hagan realidad sus visiones digitales (Joomla! Documentation, 2016).

TYPO3

Es un potente software libre que incluye útiles opciones y numerosas funcionalidades por defecto que pueden resultar muy útiles; un avanzado CMS que puede ser una excelente opción para un proyecto web y que, además, se puede instalar de manera sencilla desde el Panel de Control de Cloudbuilder Next.

TYPO3 es gratuito y de código abierto, publicado bajo licencia GNU, así que es apto para cualquier tipo de uso. Al estar desarrollado bajo PHP es capaz de ejecutarse en múltiples sistemas operativos o servidores web como Apache, NGINX o IIS. Puede trabajar con la base de datos MySQL pero también con otras más potentes como Oracle o PostgreSQL. Está perfectamente localizado en más de 50 idiomas, así que se puede utilizar sin requerir una traducción propia y, por supuesto, también es completamente válido para sitios multi-idiomias (Galán, 2016).

Contao

El joven sistema de gestión de contenidos, se ha establecido en el mercado como un sistema alternativo a otros más veteranos y completos como TYPO3 o Joomla por su flexibilidad y accesibilidad. Este gestor de contenidos de código abierto, muy fácil de usar también para principiantes, destaca, además, por la gran cantidad de extensiones que permiten ampliar sus funcionalidades y por su adaptabilidad a páginas responsivas y en varios idiomas.

Contao es un CMS basado en la página del navegador, por lo que ya en la base se diferencia de otras soluciones como Joomla o Drupal. Su estructura jerarquizada facilita la gestión de varias páginas. A partir de una estructura básica se van fijando todas las configuraciones generales como metatags y ajustes para la redirección y la visibilidad. Gracias a módulos, hojas de estilo, plantillas, PHP y múltiples extensiones, el usuario dispone de diversas modalidades de diseño para ajustar el sistema de gestión de contenidos a sus necesidades (1and1, 2016).

WordPress

Es un CMS enfocado a la creación de blogs, desarrollado bajo la Licencia Pública General GNU/GPL la cual está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Puede publicar contenido y administrar su sitio web o blog de forma segura, rápida y sencilla, desde cualquier navegador, mediante correo electrónico, o utilizando las aplicaciones soportadas para dispositivos con sistemas operativos como Windows Phone 7, iOS (iPad, iPhone), Android, BlackBerry y Nokia Phone (Scholarium, 2017).

1.4 Drupal

Drupal es un sistema de gestión de contenidos de código abierto que sirve para administrar recursos web. Es un sistema multiusuario, multiplataforma, multilenguaje, extensible y modular. Introduce el concepto de nodo como sinónimo de tipos de contenido, cualquier recurso que se ingrese al sistema pasa a ser un nodo, que puede ser variable e incluir artículos, historias, posts, encuestas, imágenes, libros colaborativos, reseñas, recetas y más. Este nuevo concepto permite estandarizar la información asignándoles las mismas características a distintos tipos de objetos y la posibilidad de tener toda la información centralizada y a la vez catalogada. La clasificación de los nodos es un tanto arbitraria y está al servicio de los usuarios/administradores de contenidos que pueden construir los que necesite (Canez, 2016).

Drupal presenta un sistema de taxonomías robusto y potente, el contenido se organiza en categorías que se arman a través del módulo taxonomías, el cual permite generar vocabularios controlados con términos que pueden ordenarse jerárquicamente y asociarse a un tipo de nodo en particular si hiciera falta. Se puede configurar que un nodo pueda ser clasificado bajo uno o múltiples términos de un vocabulario, flexibilidad que aprecian los administradores de sistemas de contenidos rígidos (Canez, 2016).

1.5 Funcionalidades de Drupal

Drupal presenta un grupo de funcionalidades en las que se pueden encontrar:

1. Contenido flexible

Puede definir campos personalizados que podrán ser utilizados en tipos de contenido, usuarios, comentarios, términos y otras entidades. Puede almacenar los datos de esos campos en SQL, NoSQL o utilizar almacenamiento remoto.

2. Accesible

Las pantallas de administración son ahora mucho más accesibles. Las abundantes mejoras en la interfaz le facilitan la construcción de páginas web altamente accesibles.

3. Imágenes y ficheros

El soporte de imágenes en el contenido está ahora incorporado en el núcleo. Genera versiones diferentes para miniaturas, vistas previas y otros estilos de imágenes y además utiliza las gestiones privada y pública de ficheros al mismo tiempo.

4. Pruebas automáticas del código

Un nuevo entorno de pruebas automatizadas, con más de 30.000 pruebas incluidas permite el *testing* de integración continua de todos los parches al núcleo de Drupal y a los módulos contribuidos.

5. Soporte de base de datos mejorado

Una nueva capa de abstracción de base de datos provee soporte para SQLite, MySQL/MariaDB y PostgreSQL out-of-the-box. Puede instalar módulos contribuidos para utilizar MS SQL Server, Oracle, y más.

6. Mejor soporte para distribuciones

Utiliza perfiles de instalación para distribuir el producto personalizado basado en Drupal. Presenta una nueva API y configuración exportable la cual le permite capturar más opciones en código.

7. Extender

Debido a un enorme esfuerzo de la comunidad, más de 800 módulos están disponibles o bajo desarrollo activo para Drupal, incluyendo Views, Pathauto, y WYSIWYG, con muchos otros módulos actualizándose cada día (Drupal).

1.6 Modularización del crecimiento

Si bien las características básicas de Drupal son suficientes como para lanzar un sitio completo, incluye: administración de usuarios, páginas, vocabulario, comentarios y sindicación. Su potencialidad reside en su estructura modular que permite ser extendida y personalizada para distintas funciones por medio de módulos de fácil instalación. Entonces este multifacético CMS permitiría armar desde un blog personal hasta una intranet corporativa, pasando por bibliotecas digitales, wikis, sitios de comercio electrónico, álbumes de fotos y hasta recetas de cocina. Estos módulos son desarrollados por una gran comunidad que colabora para el crecimiento de esta herramienta. Algunos de los más destacados y que se los puede encontrar en la sección módulos de Drupal son:

- AdSense: permite mostrar fácilmente el servicio de publicidad de Google "AdSense"
- Book Review: sirve para publicar reseñas de libros, como si fuera un nodo más, ya que provee campos específicos para esta tarea
- Captcha: añade al sistema de registro de usuarios una validación manual (copiar el código de una imagen distorsionada) para evitar el uso masivo por parte de robots, con fines inescrupulosos.
- Daily: ofrece un nuevo tipo de nodo que sirve para mostrar tiras cómicas o frases diarias. El nodo está asociado a una fecha y esto permite su navegación a ediciones pasadas a través de un calendario (Erazo, 2012).

Otras funcionalidades.

- URL alias: permite la asignación de direcciones legibles por buscadores (que se traduce en mejor ranking) y por humanos para mayor usabilidad al ser más sencillos de recordar.
- Fácilmente traducible a otros idiomas y soporte multilingüe.
- Instalación centralizada para múltiples sitios (hasta de distintos formatos).
- Cumple con los estándares de XHTML del W3C.
- Cuenta con una herramienta de foro.
- Soporte de blogger API para publicación remota.
- Estadísticas de referencias, popularidad del contenido, entre otras.
- Encuestas online mediante uno de sus módulos.
- Alimentación RSS.
- Autorregulación de tráfico, permite configurar la desactivación de módulos de forma automática cuando hay mucho tráfico.
- Libro colaborativo como un wiki, permite la creación de un proyecto en donde muchos usuarios contribuyen en su construcción.
- Agregador de noticias para incluir enlaces a contenidos de otros sitios. Con sistema de cache y configuración de los tiempos.
- Personalización de temas: brinda la base para ampliar y personalizar las plantillas del sitio (Erazo, 2012).

1.7 Módulos de Drupal 7

El estudio de los módulos que puedan resolver el problema de la investigación se llevó a cabo mediante la instalación y el análisis de algunos de los módulos disponibles en Drupal 7 que contribuyen a la administración de contenido, entre ellos se encuentran Charts, Easychart, Highcharts y Quant de los cuales se abordará a continuación.

Charts

Charts transforma datos en información utilizando gráficas, las cuales son una forma sencilla para visualizar una buena cantidad de datos que puedan ser rápidamente entendidos.

Características

- Tipos de gráficas que soporta: pastel, línea, barra, columna, área o gráficas de Dispersión.
- Consta de una Interfaz de programación de aplicaciones (*Application Programming Interface* o API): presenta una sintaxis amigable para crear gráficas fácilmente sin tener en cuenta la biblioteca subyacente.
- Integración del módulo Views: Exhibe los datos de las vistas realizada con del módulo Views mediante gráficas.
- Brinda ejemplos: El módulo incluye varias gráficas de ejemplo en las que el usuario tiene una guía para construir sus propias gráficas o biblioteca de gráficas.
- Soporta bibliotecas de gráficos: Se pueden utilizar módulos cómo Google Charts, Highcharts, o C3 como bibliotecas de gráficos (quicksketch, 2008).

Easychart

Es una interfaz gráfica del usuario construida sobre la biblioteca de Highcharts-Javascript.

Este módulo es una herramienta especialmente para editores de contenidos, transformando fácilmente datos en formato CSV en gráficos Highcharts usando plugins de Easychart.

Características

- Define un nuevo tipo de contenido denominado 'Chart' que permite mostrar gráficas en un sitio web.
- Crea gráficas que pueden usarse en los tipos de contenidos propios.

- Se integra con el módulo WYSIWYG y provee un botón de edición que le permite añadirle gráficas al contenido WYSIWYG del usuario (los módulos CKeditor y TinyMCE son soportados).
- Presenta una interfaz amigable que permite configurar gráficas mediante los plugins de Easychart (jyve, 2014).

Highcharts

Crea gráficas en Drupal usando la biblioteca Highcharts JavaScript. Este módulo incluye una API e integración del módulo Views.

Características

- Contiene un módulo el `views_highcharts` que permite a los usuarios crear una presentación de las vistas Highchart con un número limitado de tipos de la gráfica y opciones.
- El módulo Views es independiente de las API de Highcharts.
- Presenta ejemplos de código añadidos en el documento de texto README.
- La descarga de la biblioteca Highcharts es con Drush.
- Presenta una API: el módulo Highcharts es una API de Drupal para integrarse con la biblioteca de JavaScript Highcharts. Incluye funciones como: cargar archivos de la biblioteca, inicializar y suministra múltiples gráficos de Highchart.
- Integración del módulo Views: Highcharts es un módulo incluido que usa el módulo Views como un proveedor de datos e interfaz de usuario para crear gráficos Highcharts (scottrigby, 2010).

Quant

Quant provee un motor para producir análisis cuantitativos basados en tiempos para cualquier componente Drupal. Quant toma datos sin procesar sobre las acciones normales de Drupal, como la creación de nodos, y muestran la actividad con el paso del tiempo, pudiendo configurar el tiempo que se desea analizar.

Características

- Módulos recomendados: Chart API. Provee la opción para mostrar gráficas usando las API de gráficos de Google.
- Los gráficos que provee son sobre:
 - ✓ Creación de contenidos
 - ✓ Creación de comentario

- ✓ Creación de contenidos por el tipo
- ✓ Total de contenidos creados
- ✓ Creación de usuarios
- ✓ Páginas visitadas (requiere el módulo Statistics)
- ✓ El punto de transacciones del usuario (requiere el módulo User Points)
- ✓ Creación de grupos (requiere el módulo Organic groups)
- ✓ Grupos asociados (requiere el módulo Organic groups)
- ✓ Invitaciones enviadas (requiere el módulo Invite)
- ✓ Invitaciones aceptadas (requiere el módulo Invite)
- ✓ Estatus compartidos (requiere el módulo Facebook-style statuses)
- ✓ Mensajes privados (requiere el módulo Private message)
- Provee plugins para gráficos: Tablas HTML o imágenes de gráficos (requiere el módulo Charts) usando las API de gráficos de Google.
- La API de Quant: Generar gráficas con poca información disponible al presentar una simple, pero flexible API que permite a los desarrolladores incluir gráficos a sus módulos (mstef, 2010).

Se pudo observar a través del análisis de los módulos anteriores y el estudio de funcionalidades propias de Drupal como son las estadísticas de referencias, de popularidad del contenido, entre otras, que no existe para Drupal 7 un módulo que pueda dar solución al problema de la investigación. Con el objetivo de dar respuesta a este problema se hace necesario implementar un módulo desarrollado para Drupal que tenga las funcionalidades necesarias para darle solución al problema de la investigación.

Al no encontrar un módulo que pueda darle solución al problema se procedió con la instalación de los CMS más utilizados antes mencionados, lo que permitió la identificación de características asociadas a la administración de contenido. A consideración de la autora, de las diferentes características que presentan los CMS para la administración de contenido se escogieron las más importantes para el desarrollo del módulo a implementar. La siguiente tabla muestra los CMS y las características que estos presentan.

Criterios	CMS				
	Joomla!	WordPress	TYPO3	Contao	Drupal
Filtrar contenido	Si	Si	Si	Si	Si
Establecer la cantidad de elementos que se van a mostrar al realizar la búsqueda	Si	No	No	No	No
Mostrar contenidos propios	No	Si	Si	No	No
Mostrar estadísticas sobre los contenidos creados	Si	Si	Si	Si	Si
Notificaciones automática por correo electrónico de los cambios que se vayan realizando en los contenidos	No	No	No	No	No

1.8 Herramientas utilizadas para al desarrollo

Las herramientas son productos que dan soporte a una tarea concreta dentro de las actividades de desarrollo de software. Para la realización de un módulo desarrollado con el CMS Drupal se necesita disponer de un Sistema de Base de Datos (SBD), un Sistema de Gestión de Base de Datos(SGBD), un Entorno de Desarrollo Integrado (IDE), un servidor web, una herramienta de Ingeniería de Software Asistida por Computadora (CASE) y una herramienta que permita realizarle pruebas al software. A continuación, se describen las herramientas que deben ser utilizadas:

- **PostgreSQL – SBD**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Martinez, 2010).

Para la implementación del módulo se usará PostgreSQL pero una vez terminado se puede utilizar con cualquier gestor de base de datos que permita el CMS.

- **Pgadmin3 – SGBD**

PgAdmin 3 es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados (EnterpriseDB Postgres Plus Advanced Server, Greenplum Database). Incluye:

- Interfaz administrativa gráfica
- Herramienta de consulta SQL
- Editor de código procedural
- Agente de planificación SQL
- Administración de Slony-I

PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas.

La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris.

PgAdmin III soporta versiones de servidores 7.3 y superiores, para versiones anteriores a la 7.3 se debe usar pgAdmin II (Reingart, 2016).

- **Eclipse – IDE**

Para realizar la implementación del módulo se utilizará Eclipse como plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plug-ins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. No presenta un lenguaje específico, sino que es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje Java usando el plug-ins JDT que viene incluido en la distribución estándar del IDE.

Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones (calendamaia, 2014).

- **Apache2.4.10 - Servidor web**

Es un software libre y un proyecto de la Fundación de Software Apache, con el objetivo de suministrar un servidor seguro, eficiente, y extensible que proporcione servicios HTTP en sincronía con los estándares HTTP actuales.

Características

- Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.
- Multiplataforma.
- Modular: puede ser adaptado a diferentes entornos y necesidades con los diferentes módulos de apoyo que proporciona, y presenta una API de programación de módulos para el desarrollo de módulos específicos.
- Extensible: al ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor (Opensuse, 2011).

- **Visual Paradigm – (CASE)**

Para el modelado de los diagramas se utiliza el Visual Paradigm: herramienta multiplataforma de gran alcance y fácil de utilizar. La misma proporciona un conjunto de beneficios para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (Diaz, 2013).

- **Acunetix - Pruebas de seguridad**

Hasta un 70% de los sitios web tienen vulnerabilidades que pueden conducir al robo de datos corporativos confidenciales, como información de tarjetas de crédito y listas de clientes.

Acunetix ha sido pionera en la aplicación web de tecnología de análisis de seguridad. Sus ingenieros se han centrado en la seguridad web desde 1997 y ha desarrollado una ingeniería líder en el análisis de sitios web y detección de vulnerabilidades.

Acunetix web vulnerability scanner incluye características innovadoras como:

- Tecnología AcuSensor.
- Un cliente de analizador automático de secuencia de comandos que permite realizar pruebas de seguridad de Ajax y aplicaciones Web 2.0.
- Las herramientas de testeado de inyección SQL y de *Cross-site scripting*⁵ más avanzadas y profundas de la industria.
- Herramientas avanzadas de penetración, como HTTP Editor y HTTP Fuzzer.
- Herramientas para fácil aseguramiento de formularios web y contraseñas.
- Soporte para páginas con CAPTCHA, single sign-on y mecanismos con factor de autenticación.
- Facilidad de generación de informes amplios, incluyendo informes de cumplimiento PCI.
- El escaneo inteligente detecta el tipo de servidor web y lenguaje de la aplicación.
- Acunetix escanea y analiza sitios web incluyendo contenido flash, SOAP y AJAX.
- Permite explorar un servidor web y ejecutar comprobaciones de seguridad contra los servicios de red que se ejecutan en el servidor (Seaq).

1.9 Lenguajes de programación

Los lenguajes utilizados en el desarrollo del módulo están divididos en dos grupos los que se encuentran del lado de cliente y los del lado de servidor, a continuación se describen los lenguajes usados en cada caso.

Lenguajes de programación del lado del cliente

- Javascript

⁵ Es un tipo de inseguridad informática o agujero de seguridad típico de las aplicaciones Web, que permite a una tercera parte inyectar código JavaScript en páginas web vistas por el usuario, evitando medidas de control.

Es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto se puede decir que nació como un *"lenguaje de scripting"* del lado del cliente, sin embargo, hoy Javascript es mucho más. Las necesidades de las aplicaciones web modernas y el HTML5 han provocado que el uso de Javascript haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel (Desarrolloweb).

Javascript se utiliza además para la manipulación de algunos eventos del lado de cliente cuando los usuarios utilizan las funcionalidades del módulo.

- **CSS**

Para el diseño de las interfaces se utilizará CSS (hojas de estilo en cascada) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML permitiendo crear páginas web de una manera más exacta. Logra personalizar la página al gusto del usuario, permite incluir márgenes, tipos de letra, fondos, colores, entre otros, que no se podía hacer utilizando HTML (Lazaro, 2001).

- **HTML**

Para estructurar el contenido de las páginas web se utiliza HTML. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, entre otros (Alvarez, 2001).

Lenguajes de programación del lado del servidor

- **PHP**

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores (Alvarez, 2001).

1.10 Tecnologías utilizadas para al desarrollo

Ajax (JavaScript asíncrono y XML)

Ajax no es más que una forma de programar aplicaciones interactivas para Web, no es una tecnología, propiamente dicho sino la unión de varias tecnologías que se desarrollan de forma autónoma, que se unen de formas nuevas y sorprendentes.

Las tecnologías que forman parte de AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías (Pérez, 2008).

JQuery

Es una biblioteca de lenguaje Javascript, facilita el trabajo para programar en este lenguaje. Ofrece una infraestructura con la que se tiene mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Con jQuery se obtiene ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax (Chaffer, 2016).

Una de las características y ventajas que ofrece Drupal es que ésta versión de Drupal se distribuye con las versiones jQuery 1.4.4 y jQuery UI 1.8.7, las cuales vienen integradas en su núcleo, convirtiéndose en la librería oficial para desarrollo de aplicaciones del lado del cliente con el CMS. Esta ventaja de integración y compatibilidad hace ser a JQuery la librería a usar para el trabajo en el lado del cliente.

1.11 Metodología de desarrollo

Los procesos ágiles de desarrollo de software, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. El software es desarrollado en una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar

demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto (Ingeniería de Software, 2016).

Proceso Unificado Ágil (AUP)

Es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo desarrollo dirigido por pruebas, modelado ágil, gestión de cambios ágil, y refactorización de base de datos para mejorar la productividad.

El AUP establece un modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP (Ingeniería de Software, 2016).

Para guiar el proceso de desarrollo se decide utilizar una variación de la metodología AUP, ya que está especialmente indicada a grupos de trabajo pequeños, fomenta el trabajo en equipo y se centra en la naturaleza colaborativa de desarrollo de software, dicha metodología es AUP-UCI por estar diseñada para el ciclo de vida definido para la actividad productiva de la UCI.

Ciclo de Vida de la metodología AUP-UCI

En AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados:

- **Concepción:** El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- **Transición:** el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción (Ingeniería de Software, 2016).

Principios de la AUP-UCI

La AUP es ágil, porque está basada en los siguientes principios:

1. El personal se encuentra capacitado pero algunos quieren una orientación de alto nivel y / o formación, por lo que el producto proporciona enlaces a muchos de los detalles si alguien del personal se encuentra interesado, pero no obliga a aquellos que no lo deseen.
2. Simplicidad. Todo se describe concisamente utilizando un pequeño conjunto de páginas.
3. Agilidad. El ajuste a los valores y principios de la Alianza Ágil.
4. Centrarse en actividades de alto valor. La atención se centra en las actividades que son esenciales para el desarrollo, no todas las actividades que suceden forman parte del proyecto.
5. Herramienta de la independencia. Se puede usar cualquier conjunto de herramientas con AUP. Lo aconsejable es utilizar las herramientas que son las más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.
6. Adaptación del producto desarrollado utilizando la metodología AUP para satisfacer las propias necesidades del usuario es de fácil acomodo común a través de cualquier herramienta de edición de HTML. No se necesita comprar una herramienta especial, o tomar un curso, para adaptar la AUP (Ingeniería de Software, 2016).

1.12 Conclusiones

Los conceptos asociados al objeto de estudio, el análisis de distintos CMS además de los módulos encargados de la administración de contenidos disponibles en Drupal 7 sirvieron para formar una base y guía para el desarrollo del módulo.

El estudio de algunos de los CMS más utilizados dígame: WordPress, Joomla!, TYPO3 y Contao permitió identificar determinadas características de la administración de contenidos que permiten una mejor gestión de los mismos.

Con la selección de las herramientas, tecnologías y lenguajes de programación se logró obtener una base tecnológica idónea que permitirá el desarrollo de la solución.

Capítulo 2. Módulo para el CMS Drupal 7.

En este capítulo se brinda una propuesta general del módulo a desarrollar. Se realiza un levantamiento de requisitos funcionales y no funcionales a partir del modelo de dominio construido basándose en los objetos relevantes del problema, además se describe cada requisito para guiar el proceso de desarrollo de software.

2.1 Descripción de la propuesta de solución

Como propuesta para darle solución al problema de la presente investigación, se propone el desarrollo de un módulo que permita facilitar la gestión de contenido a los administradores y usuarios de los sitios. Para ello este módulo debe:

- Filtrar contenido por título, autor, tipo, idioma, estado, y fecha de creación.
- Establecer la cantidad de elementos que se van a mostrar al realizar la búsqueda
- Mostrar contenidos propios
- Mostrar estadísticas sobre el total de contenidos creados y contenidos propios mediante tablas o gráficos.
- Enviar a los usuarios por correo electrónico notificaciones al insertar, eliminar o modificar contenidos de su autoría.
- Enviar al administrador por correo electrónico notificaciones cuando un usuario ha insertado un contenido.

2.2 Modelo de dominio

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los eventos que suceden en el entorno en el que trabaja el sistema. El modelo de dominio se describe mediante diagramas UML (especialmente mediante diagramas de clases). De manera general el modelo del dominio ayuda a los usuarios, clientes, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se sitúa el sistema (Larman, 2000).

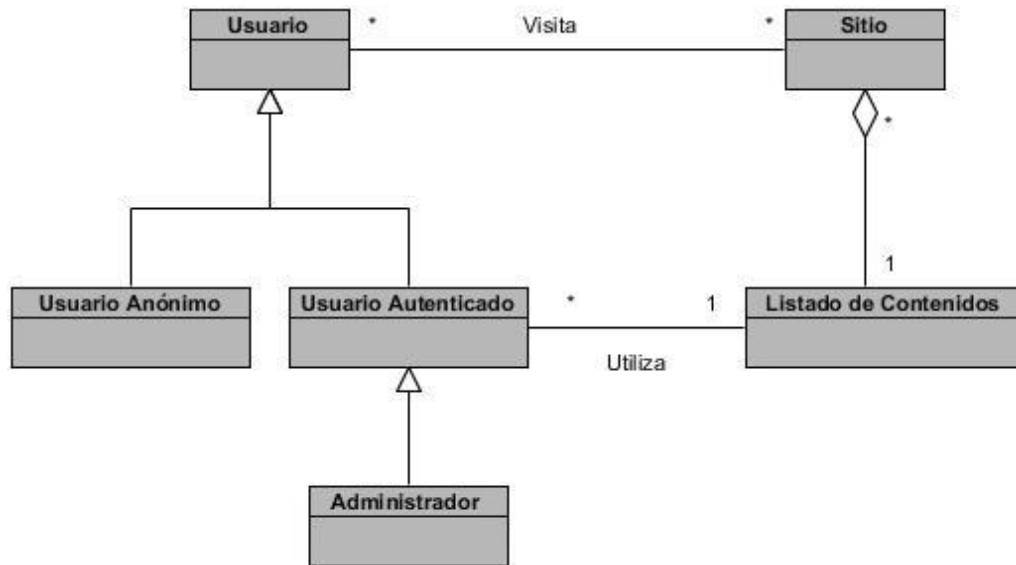


Figura 1: Modelo de dominio.

Sitio: Donde se ejecuta el CMS Drupal.

Lista de contenidos: Listado de los contenidos de los sitios web existentes en CMS Drupal.

Usuario: Persona que visita el sitio de Drupal.

Las clases Usuarios autenticado, Usuarios anónimos y Administrados son tipos de usuarios que se conectan al sitio y en dependencia del tipo son las responsabilidades asociadas, que se describen a continuación.

2.3 Roles y responsabilidades

La tabla muestra los roles que Drupal dispone inicialmente para sus usuario, así como una breve descripción de sus responsabilidades y niveles de acceso (Rodríguez, 2011).

Tabla 1: Roles y responsabilidades.

Rol	Descripción
Administrator (Administrador)	Este rol está asociado al usuario administrador del sistema. Drupal asignará a este rol todos los permisos disponibles, tanto de los módulos del núcleo como de los módulos adicionales

	que se instalen en el sitio. Este rol puede editarse e incluso eliminarse, aunque no es recomendable.
Usuario registrado	Este rol es el que se asigna automáticamente a todos los usuarios que disponen de una cuenta y han iniciado una sesión en el sitio con dicha cuenta. Por tanto, todos los usuarios registrados tendrán como mínimo este rol y además podrán tener asignados otros roles adicionales. Este rol tampoco puede ser eliminado del sistema.
Usuario anónimo	Este rol se emplea para los usuarios que no tienen cuenta o que no se han autenticado (visitantes en general). No puede ser eliminado del sistema.

Además de los roles por defecto, es posible crear nuevos roles para diferenciar las tareas que pueden realizar los usuarios. Es conveniente definir un conjunto de roles lo suficientemente descriptivos como para que, sólo conociendo los roles que un usuario tiene asignado, se pueda deducir qué acciones podrá hacer en el sitio (Rodríguez, 2011).

2.4 Requisitos funcionales

Los requisitos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir. Expresan una especificación detallada de las responsabilidades del sistema en cuestión y permiten determinar, de una manera clara, lo que debe hacer el sistema (Andrés, 2005).

El objetivo fundamental de los RF es identificar y documentar lo que se espera del sistema, de una forma clara y comprensible para el cliente y los miembros del equipo de desarrollo. El reto consiste en definirlos de manera inequívoca, de modo que se detecten los riesgos y no se presenten sorpresas al momento de entregar el producto (Andrés, 2005).

Para el módulo se determinaron 15 RF los cuales se muestran a continuación:

- RF1- Filtrar contenidos por título.
- RF2- Filtrar contenidos por fecha de creación.
- RF3- Filtrar contenidos por tipo.
- RF4- Filtrar contenidos por autor.
- RF5- Filtrar contenidos por estado.
- RF6- Filtrar contenidos por idioma.
- RF7- Definir cantidad de elementos a mostrar.
- RF8- Mostrar gráfica de las estadísticas de los contenidos propios.
- RF9- Mostrar gráfica de las estadísticas de los contenidos existentes en el sistema.
- RF10- Mostrar gráfica de las estadísticas de los contenidos según el idioma.
- RF11- Mostrar gráfica de las estadísticas de los contenidos según su estado.
- RF12- Mostrar listado de los contenidos de usuario.
- RF13- Enviar notificaciones al usuario por correo electrónico cuando un contenido ha sido creado.
- RF14- Enviar notificaciones al usuario por electrónico cuando un contenido ha sido modificado.
- RF15- Enviar notificaciones al usuario por electrónico cuando un contenido ha sido eliminado.

2.5 Requisitos no funcionales

Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener, aunque no formen parte de su función. Existen múltiples categorías para clasificar los requisitos no funcionales, siendo las siguientes las más representativas, aunque no limitan la definición de otras (Andrés, 2005).

Para el módulo se definieron los siguientes RNF:

Usabilidad

- RNF1 El módulo debe presentar un acceso fácil y rápido, para facilitar el uso del mismo por usuarios con conocimientos básicos en el campo de la informática.
- RNF2 La interfaz debe ser sencilla y amigable de manera que potencie la comodidad del usuario para su trabajo, además de que las opciones más usadas presentarán vías rápidas y cómodas de invocarse.

Diseño e implementación

- RNF3 Sistema de Gestión de Contenidos (CMS): El módulo se desarrollará sobre Drupal en su versión 7.

Seguridad

-RNF4 El módulo, así como la información que este brinda, se encontrarán protegidos contra accesos no autorizados utilizando mecanismos de seguridad propios del Drupal.

2.6 Historias de usuario

La Historia de Usuario (HU) describe cómo se realizará un determinado requisito en el sistema. En otras palabras, el objetivo que persigue una HU es detallar una funcionalidad que del módulo a desarrollar para mostrar una clara visión al programador.

Tabla 2: HU Filtrar contenido por título.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Filtrar contenidos por título.
Modificación de Historia de Usuario Número: 1	
Usuario: Anabel García García	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 días.
Riesgo en Desarrollo: Alto	Puntos Reales: 2 días.
Descripción: La HU inicia cuando el usuario después de autenticado desea buscar un contenido específico por el título. El usuario accede al bloque de administración, ubicado en la parte superior del portal, selecciona la opción: Contenidos . El sistema le muestra una interfaz con los criterios disponibles para filtrar el contenido. El usuario selecciona el campo de texto con el parámetro Título y una vez introducido el título que desea buscar y haber seleccionado el botón: Aplicar , el sistema le mostrará una lista con los contenidos que tengan ese título, y entonces habrá culminado la HU.	
Observaciones: 1- El usuario introduce un título incorrecto y al no existir el sistema no le devuelve ningún contenido.	

Interfaz:

Panel de control **Content** Estructura Apariencia Usuarios Módulos Configuración Informes Ayuda Cerrar sesión
Bienvenido, root

Lista de contenidos **CONTENIDO** COMENTARIOS

+ Agregar contenido

Título

Tipo - Seleccione -

Publicado - Seleccione -

Idioma - Seleccione -

Autor

Separar por coma la lista de usuarios

Creación a partir del

Ejemplo: 06/20/2017

hasta el

Ejemplo: 06/20/2017

Elementos 10 **Filtrar**

OPERACIONES
- Seleccione una operación - **Ejecutar**

Seleccione algún filtro y haga click en Filtrar para ver los resultados

Tabla 3: HU Definir cantidad de elementos a mostrar.

Historia de Usuario	
Número: HU_7	Nombre Historia de Usuario: Definir cantidad de elementos a mostrar.
Modificación de Historia de Usuario Número: 1	
Usuario: Anabel García García	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 días.
Riesgo en Desarrollo: Alto	Puntos Reales: 2 días.
Descripción: La HU inicia cuando el usuario después de autenticado desea definir la cantidad de elementos a mostrar. El usuario accede al bloque de administración, ubicado en la parte superior del portal, selecciona la opción: Contenidos .	

El sistema le muestra una interfaz con los criterios disponibles para filtrar el contenido. El usuario despliega el parámetro **Elementos** y el sistema les mostrará las opciones 5, 10, 20, 40 60, una vez elegida la cantidad de elementos que desea mostrar y haber seleccionado el botón: **Aplicar**, el sistema le mostrará una lista con la cantidad de elementos que el usuario seleccionó, y entonces habrá culminado la HU.

Observaciones:

1-El usuario elije una cantidad de elementos y es menor a la cantidad de contenidos encontrados, el sistema le permite pasar la página para ver los otros contenidos encontrados.

2--El usuario elije una cantidad de elementos y es mayor a la cantidad de contenidos encontrados, el sistema le va a mostrar un listado de contenidos con menos elementos que la cantidad elegida por el usuario.

Interfaz:

The screenshot shows a web application interface for content management. At the top, there is a navigation bar with the following items: Panel de control, Content (selected), Estructura, Apariencia, Usuarios, Módulos, Configuración, Informes, Ayuda, and Cerrar sesión. Below the navigation bar, there is a header area with 'Lista de contenidos' and two tabs: 'CONTENIDO' (selected) and 'COMENTARIOS'. The main content area is titled 'Agregar contenido' and contains a form with the following fields and controls:

- Título:** A text input field.
- Tipo:** A dropdown menu with the text '- Seleccione -'.
- Publicado:** A dropdown menu with the text '- Seleccione -'.
- Idioma:** A dropdown menu with the text '- Seleccione -'.
- Autor:** A text input field with a search icon on the right. Below it, the text 'Separar por coma la lista de usuarios' is displayed.
- Creación a partir del:** A text input field with the example 'Ejemplo: 06/20/2017' below it.
- hasta el:** A text input field with the example 'Ejemplo: 06/20/2017' below it.
- Elementos:** A dropdown menu with the value '10' selected.
- Filtrar:** A button to apply the filters.
- OPERACIONES:** A dropdown menu with the text '- Seleccione una operación -' and an 'Ejecutar' button.

At the bottom of the form, there is a message: 'Seleccione algún filtro y haga click en Filtrar para ver los resultados'.

Tabla 4: HU Mostrar notificaciones de usuario cuando un contenido ha sido modificado.

Historia de Usuario

Número: HU_14	Nombre Historia de Usuario: Mostrar notificaciones de usuario cuando un contenido ha sido modificado.
Modificación de Historia de Usuario Número: 1	
Usuario: Anabel García García	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 días.
Riesgo en Desarrollo: Alto	Puntos Reales: 2 días.
<p>Descripción: La HU inicia cuando el usuario después de autenticado desee modificar algún contenido. En la página principal del sistema después de identificar el contenido que quiere modificar selecciona la opción Editar se muestra una interfaz con varias opciones para modificar, una vez realizados los cambios en el contenido el usuario elije la opción Salvar y a su vez el sistema le envía una correo de confirmación al Autor del artículo diciendo que este ha sido modificado y entonces habrá culminado la HU.</p>	
<p>Observaciones:</p> <p>1-El usuario modifica un contenido y el sistema muestra un mensaje en la página principal diciendo que el contenido ha sido modificado.</p> <p>2-El usuario cancela la operación de editar el contenido y no se le envía correo al autor.</p>	
<p>Contenido del Correo:</p> <p>Asunto: Contenido actualizado</p> <p>Cuerpo: Se actualizó el contenido: (nombre del contenido), por el usuario: (Nombre del usuario que modificó el contenido).</p>	

2.7 Modelo de datos

El modelo de datos constituye el principal elemento de diseño de bases de datos que consiste en representar: objetos (entidades que existen y que se manipulan), atributos (características básicas de estos objetos) y relaciones (forma en que se enlazan los distintos objetos entre sí). Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos (Butcher, 2008).



Figura 2: Modelo de datos.

La tabla user_preference representa las preferencias de usuario, esta tabla se añadirá a la base de datos al instalar el módulo, constituidas por un id, el usuario y las preferencias de este. La tabla user representa los usuarios registrados en Drupal, estará constituida por varios atributos como son id, nombre, contraseña, correo, entre otros.

2.8 Arquitectura de Drupal

El módulo está implementado sobre el CMS Drupal, por lo que la arquitectura y los patrones de diseño usados se heredan del mismo. El secreto de Drupal para conseguir su reconocida flexibilidad y facilidad en la creación de sitios web es la abstracción y organización en capas que aplica en el tratamiento de los contenidos. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal

estructura los contenidos en una serie de elementos básicos. Estos son los datos (nodos), módulos, bloques y menús, permisos de usuario y plantillas (Drupal, 2017).

En la capa “Nodos” se encuentran los elementos básicos en que Drupal almacena la información, los contenidos. Así a medida que el sitio web crece, lo va haciendo el número de nodos los cuales van formando un “depósito de nodos” cada vez mayor y es donde se guardan los datos del módulo desarrollado.

Los “Módulos” son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web, al igual que el módulo desarrollado que se instalan en Drupal proporcionándole nuevas funcionalidades.

La siguiente capa de Drupal la constituyen los “Bloques y menús” presentes en el módulo para la administración de contenidos desarrollado con Drupal, permitiendo estructurar y organizar los contenidos de este como es el caso del menú de estadística y los filtros para la búsqueda de contenidos. Es decir que son los elementos que albergan y permiten que el usuario acceda a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos.

La siguiente capa importante en Drupal es la de control de usuarios y permisos. Actualmente, la mayor parte de sitios web son multiusuario, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta finalidad Drupal dispone de un registro de usuarios y de roles que permite especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario. Es decir que las operaciones que pueda realizar el usuario con el módulo se encuentran limitadas por la capa de control de usuarios y permisos de Drupal.

La última capa, es la capa de “Plantillas” y es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario, heredándola también el módulo implementado. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del sitio web sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que se quiere es renovar la apariencia de un sitio web.

A continuación, se muestra la figura que representa las capas pertenecientes a la arquitectura de Drupal.

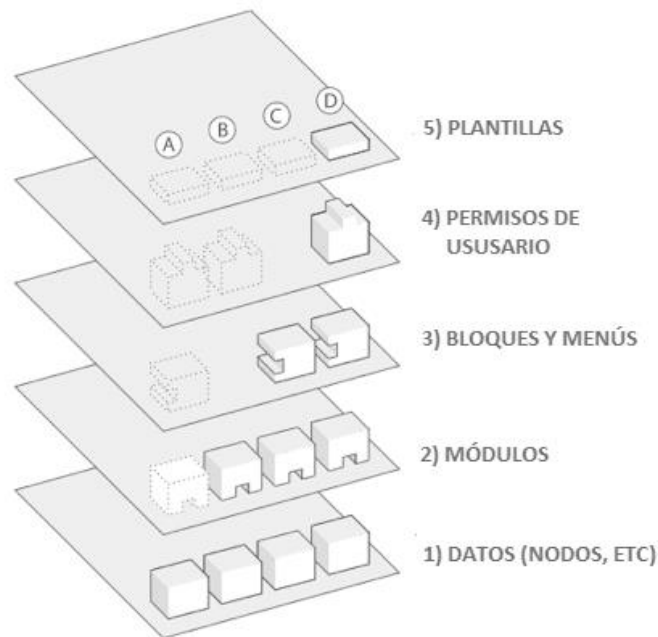


Figura 3: Arquitectura de Drupal.

2.9 Patrones de diseño

Drupal hace uso de patrones de diseño como: *singleton* (sencillo o instancia única), *decorator* (decorador), *observer* (observador), *bridge* (puente), *chain of responsibility* (cadena de responsabilidad) y *command* (comando). Estos permiten diseñar sistemas seguros y que a su vez cumplan con los estándares de diseño establecidos por normas internacionales para el desarrollo de aplicaciones web (Elvira, 2011).

Los patrones que serán usados para el desarrollo del módulo para la gestión y administración de contenidos desarrollados con Drupal se evidencian de la forma siguiente:

Observer (Observador)

Los módulos que implementan un hook⁶ determinado por evento de inserción o actualización de una determinada entidad, son declarados como observadores de dichas entidades con las que interactúan. En el módulo se implementa el hook `manage_content_node_insert`, que se ejecuta al insertar un contenido en el sitio, por lo que el módulo es declarado como observador ya que se encuentra a la espera de un evento de inserción.

Bridge (Puente)

La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón bridge. Cada módulo debe ser programado de manera que sea independiente del motor de base de datos que utiliza el sistema. Esto se logra por la capa de abstracción de base de datos, sobre la que se pueden desarrollar nuevas siguiendo la API definida, para añadir soporte para nuevos sistemas gestores de base de datos, sin necesidad de modificar el código de los módulos (Larman, 2016).

Este patrón se evidencia en el módulo `manage_content` que fue programado logrando independencia del motor de base de dato (BD) que se utilice. Esto se evidencia en los hooks `manage_content_node_insert` y `manage_content_node_delete` que permiten conectarse a la BD y realizar consultas abstrayéndose del SGBD que se emplee para manejar los datos almacenados.

Chain of responsibility (Cadena de responsabilidades)

El sistema de menús de Drupal es la evidencia del patrón *Chain of responsibility*. En cada petición de una página, el sistema de menús de Drupal determina si hay algún módulo para responder la petición, si el usuario tiene acceso al recurso solicitado y cuál función se debe llamar para procesar la petición. En este proceso se transmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos (E Camacho, 2014).

⁶ Son funciones PHP que aumentan la funcionalidad de Drupal permitiendo a los módulos interactuar con el core. De esta manera se puede alterar los comportamientos default de Drupal, y agregar nueva funcionalidad, si esos son los requerimientos para nuestro sitio.

En el módulo `manage_content` mediante el `manage_content_form_alter` en dependencia de la petición recibida por el módulo, se delega la responsabilidad a la función encargada de atenderla. De esta forma se continúa la cadena hasta que un módulo atienda la petición, hasta que un módulo deniegue el acceso o hasta que la cadena se agote.

2.10 Modelo de despliegue

Un diagrama de despliegue es un tipo de diagrama del lenguaje unificado de modelado que se utiliza para modelar el hardware utilizado en el despliegue del sistema y las relaciones entre sus componentes. Muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos (Pressman, 2010).

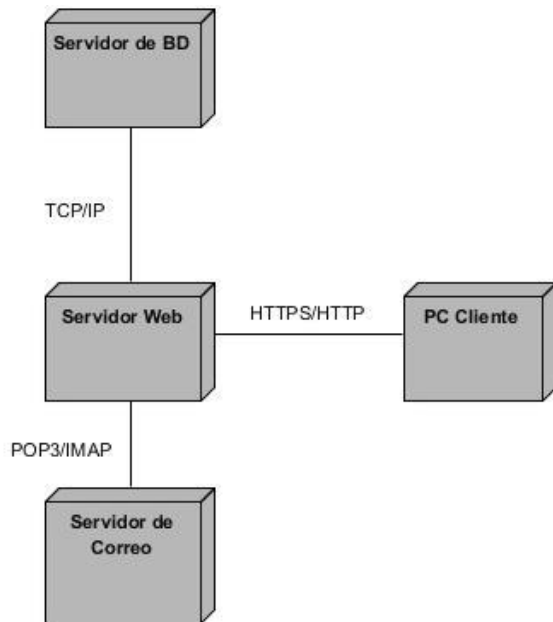


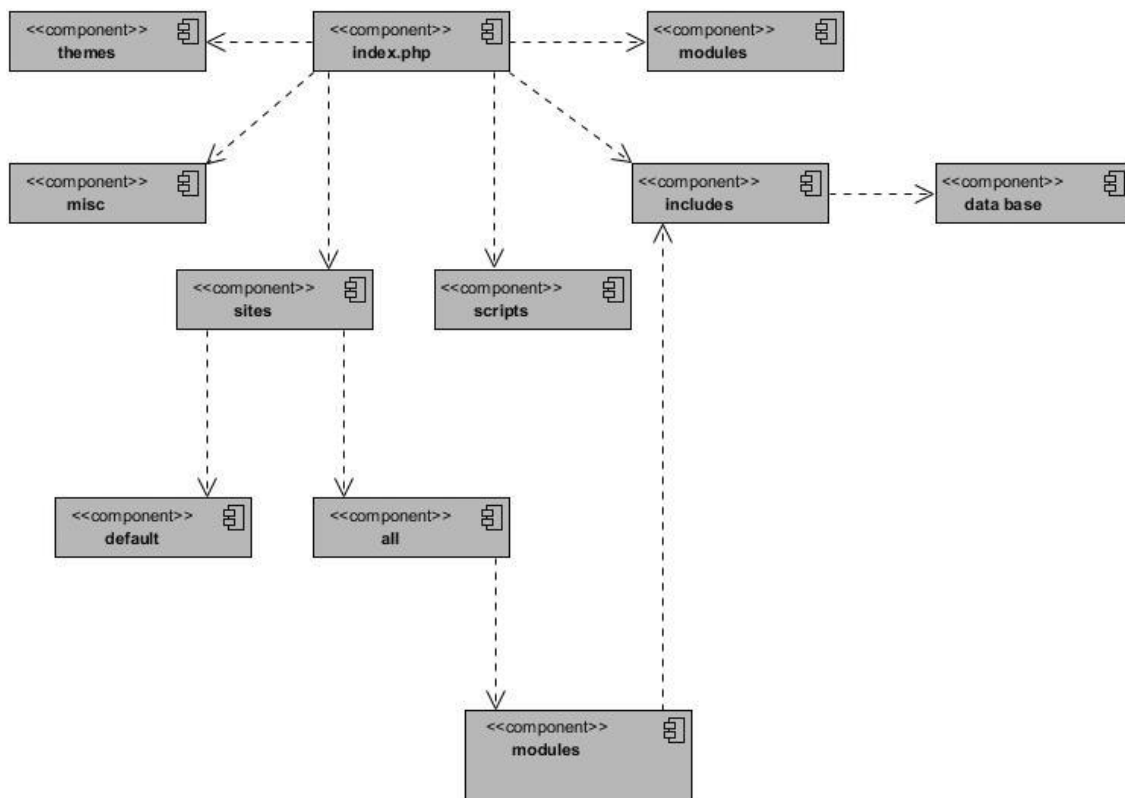
Figura 4: Modelo de despliegue.

En el diagrama anterior se puede observar que el sistema cuenta con 4 nodos principales. El nodo PC Cliente que requiere de un navegador web, el nodo Servidor Web donde debe estar instalado el CMS Drupal, el nodo Servidor de Base de Datos en el cual debe estar instalado el Sistema Gestor de Bases de Datos PostgreSQL con la base de datos perteneciente a Drupal y el nodo Servidor de Correo que presta servicio de correo electrónico. El nodo PC Cliente representa las estaciones de trabajo de los usuarios que

se conectan al CMS, las mismas realizan peticiones al Servidor Web mediante el protocolo HTTPS o HTTP. El nodo Servidor Web, a su vez, puede estar conectado al nodo Servidor de Base de Datos mediante el protocolo TCP\IP y con el nodo Servidor de Correo mediante el protocolo POP3 o el IMAP.

2.11 Diagrama de componentes

Un diagrama de componentes permite ilustrar con más facilidad la estructura general de un sistema de software. En este artefacto se muestran las organizaciones y dependencias lógicas entre los componentes del software, ya sean estos ficheros de código fuente, binarios o ejecutables. Los elementos de modelado que lo conforman son los componentes y paquetes, los que muestran la estructura de alto nivel del modelo de implementación (Giner, y otros, 1999).



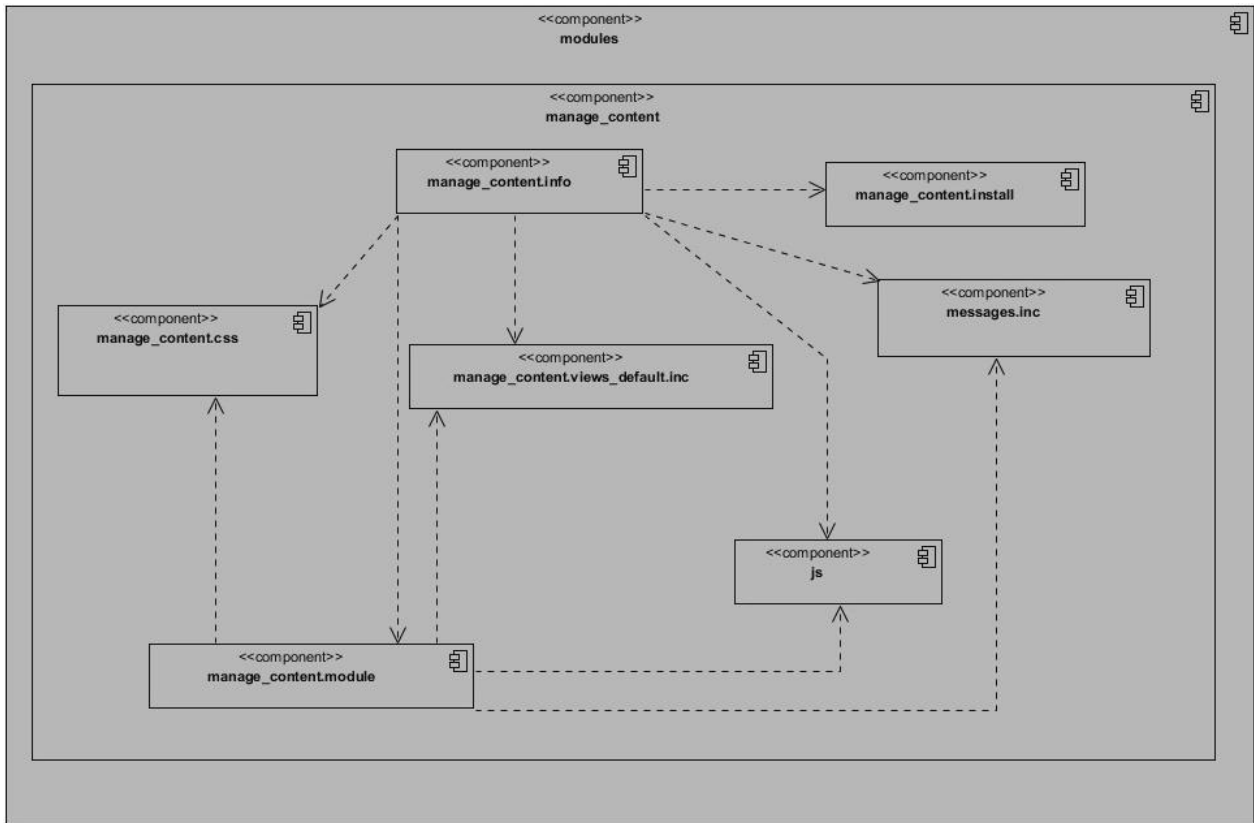


Figura 5: Diagrama de componentes.

2.12 Conclusiones

Mediante el análisis y diseño del módulo para la administración de contenidos desarrollado con Drupal 7 y la representación y descripción de los artefactos generados se garantiza una mejor comprensión del módulo a implementar.

La especificación de los requisitos funcionales y no funcionales sirve como guía y ayuda a un mejor entendimiento de los resultados que se procuran obtener de una manera precisa.

La arquitectura y los patrones de diseño identificados, permitieron establecer las bases para promover las buenas prácticas de programación.

Capítulo 3. Validación del Módulo para el CMS Drupal.

El módulo desarrollado se somete a pruebas funcionales y de integración. En este capítulo se describe este proceso, los estándares de codificación empleados durante la implementación. Además, se muestra una descripción de las pruebas seleccionadas para validar la implementación realizada y los resultados obtenidos en la ejecución de cada una de ellas

3.1 Estándares de codificación

Identación

Para que el código sea más legible se puede insertar espacios en blanco. En Drupal se usa indentar con 2 espacios, y nunca con tabulaciones, también se debe tener presente que nunca se debe dejar espacios en blanco al final de cada línea, si se desea continuar en la siguiente línea simplemente se presiona la tecla <ENTER> (Guzzo, 2016).

Estructuras de control

Las estructuras de control deben tener el siguiente formato:

```
if (condición_1 || condición_2) {
    acción_1;
}
elseif (condición_3 && condición_4) {
    acción_2;
}
else {
    acción predeterminada;
}
```

- Espacio entre las palabras reservadas (if, while, for, else, entre otros) y el paréntesis de apertura
- Entre el paréntesis de cierre y la llave de apertura { también tiene su correspondiente espacio
- Siempre se debe usar las llaves de apertura y cierre

- Las estructuras else y elseif deben ir en líneas aparte, debajo del cierre de la estructura, en este caso debajo de la llave de cierre (Guzzo, 2016).

Ejemplo de switch:

```
switch (condición) {  
    caso_1:  
        acción_1;  
        break;  
  
    caso_2:  
        acción_2;  
        break;  
  
    default:  
        acción predeterminada;  
}
```

Ejemplo de do:

```
do {  
    acción;  
} while ($condición);
```

Etiquetas de apertura y cierre de PHP

Al escribir código PHP, se debe utilizar las etiquetas de apertura `<?php` y la etiqueta de cierre `?>`, en los archivos `.module` y `.inc` se puede omitir la etiqueta de cierre siempre y cuando se trabaje solo con PHP, omitir la etiqueta de cierre evita que queden líneas en blanco lo que podría dar como resultado un error típico "Cannot modify header information - headers already sent by ..".

Cuando se trabaja con PHP puro, se puede evitar la etiqueta de cierre, pero dentro de HTML se debe utilizar la etiqueta de apertura de PHP `<?php` y la etiqueta de cierre de PHP `?>` (Guzzo, 2016).

Uso de comillas

Se puede utilizar comillas simples ('texto') o comillas dobles ("texto") para delimitar las cadenas, las comillas dobles permiten interpretar el valor de una variable, entonces para "<h1>\$title</h1>", el resultado sería el valor de la variable \$title (Guzzo, 2016).

Uso de punto y coma

En Drupal siempre se debe finalizar las líneas de código PHP con el punto y coma (Guzzo, 2016).

Operadores

Todos los operadores binarios (operadores que se utilizan entre dos valores), deben tener un espacio de separación entre ellos, esto aplica a operadores como +, -, *, /, =, ==, !=, <, >, ., .=, +=, +-, entre otros (Guzzo, 2016).

Ejemplo de una asignación:

```
$numero = 3;
```

Los operadores unarios como el ++ y el -- no deben tener separación

Ejemplo de operador unario

```
$numero++;
```

Funciones

Los nombres de las funciones deben estar escritas en minúsculas, y las palabras separadas por un guión bajo. Siempre se debe escribir como prefijo el nombre del módulo, tema, entre otros para evitar así generar nombres duplicados.

Entre el nombre de la función y el paréntesis de inicio de los argumentos no se debe dejar espacios, cada argumento debe estar separados por una coma y luego un espacio. Se debe colocar un espacio entre el cierre del paréntesis y la apertura de la llave que contiene el bloque de código (Guzzo, 2016).

Ejemplo:

```
function forum_help($path, $arg) {  
    bloque de código  
    ....  
}
```

```
}
```

La llamada a la función tiene las mismas reglas que la declaración, ejemplo:

```
<span style="color: #222222; font-family: 'Bitstream Vera Sans Mono', Monaco, 'Lucida Console',  
monospace; font-size: 12px; font-style: normal; font-variant: normal; line-height: 17.984375px; background-  
color: #f6f6f2;">$var = foo($bar, $baz, $quux);</span>
```

Cuando se asigna una serie de funciones, se puede dejar varios espacios para darle una organización visual, aquí no hay restricciones, ejemplo:

```
$bar = 'NombreDeMiClase';  
$foo = new $bar();  
$foo = new $bar($arg1, $arg2);
```

Arreglos

Dentro del arreglo o matriz, se separa con un espacio (luego de la coma) los valores. El operador => debe separarse por un espacio a ambos lados. Cuando se superes los 80 caracteres en la declaración del arreglo, se debe escribir cada uno de los elementos en una única línea, la indentación debe tener 2 espacios (Guzzo, 2016).

Ejemplos:

```
$some_array = array('hello', 'world', 'foo' => 'bar');  
  
$form['title'] = array(  
  '#type' => 'textfield',  
  '#title' => t('Title'),  
  '#size' => 60,  
  '#maxlength' => 128,  
  '#description' => t('Título del nodo.'),  
);
```

Constantes

Siempre se debe escribir en mayúsculas los nombres de las constantes, si se componen de varias palabras entonces se pueden separar con guiones bajos, al igual que en las funciones se debe colocar de prefijo el nombre del módulo o tema a donde pertenece, el objetivo es evitar crear nombres duplicados, el prefijo debe estar también en mayúsculas (Guzzo, 2016).

```
<?php
/**
 * The current system version.
 */
define('VERSION', '7.10');

/**
 * Minimum supported version of PHP.
 */
define('DRUPAL_MINIMUM_PHP', '5.2.4');
?>
```

Variables globales

Si se necesita definir variables globales se debe comenzar con el nombre con un guión bajo seguido por el nombre del módulo o tema y otro guión bajo antes del nombre de la variable (Guzzo, 2016).

```
Global $_forum_pagina_foros;
```

Nombre de los módulos

Por norma general, el nombre de un módulo nunca debe incluir guiones bajos, aunque este se componga de varias palabras. Pueden encontrarse nombres de módulos con guiones bajos ya que esta no es una regla obligatoria (Guzzo, 2016).

Ejemplo:

```
Mimodulo
```

Nombre de los archivos

Siempre se debe escribir los nombres de archivos en minúsculas, salvo los nombres de archivos que contengan documentación como por ejemplo, README.txt, INSTALL.txt, entre otros (Guzzo, 2016).

Ejemplo de URLs

Por convención, siempre que se vaya a utilizar una URL como ejemplo, esta debe ser "example.com" (Guzzo, 2016).

Idioma

Drupal utiliza el idioma inglés como base para todo el sistema, por este motivo se debe escribir la documentación (módulos, temas, archivos, nombre de funciones, variables, entre otros.) en este idioma, también las cadenas de texto que ve el visitante del sitio ya que la traducción del mismo se realiza mediante la función t() (Guzzo, 2016).

Comentar el código

Se debe diferenciar los comentarios que se utilizan para aclarar determinados bloques de código, ya que estos se pueden colocar donde se crea que sea necesario, de los comentarios de documentación que son los que se escriben al principio del archivo, o antes de declarar alguna función, estos últimos son sumamente importantes ya que por medio de una API se genera la documentación de ayuda desde las etiquetas que se emplean.

En el tipo de documentación que se usa para aclarar algún bloque de código se utiliza las etiquetas de apertura /* y de cierre */, entre ellas podemos colocar varias líneas de comentario, o se puede utilizar las etiquetas // para un comentario de una simple línea (Guzzo, 2016).

```
// comentario de una simple línea
```

```
/**  
 * Comentario de varias  
 * líneas de texto  
 */
```

```
function system_init() {  
  
    // Otro comentario de una sola línea  
    ... mi bloque de código  
}
```

3.2 Pantallas del módulo de administración de contenidos en aplicaciones web desarrolladas con Drupal

A continuación se muestran algunas imágenes de las funcionalidades e información que brinda el módulo desarrollado.

Total de los contenidos creados

Se muestra un gráfico de pastel que muestra la cantidad de contenido que se han creado de cada tipo.

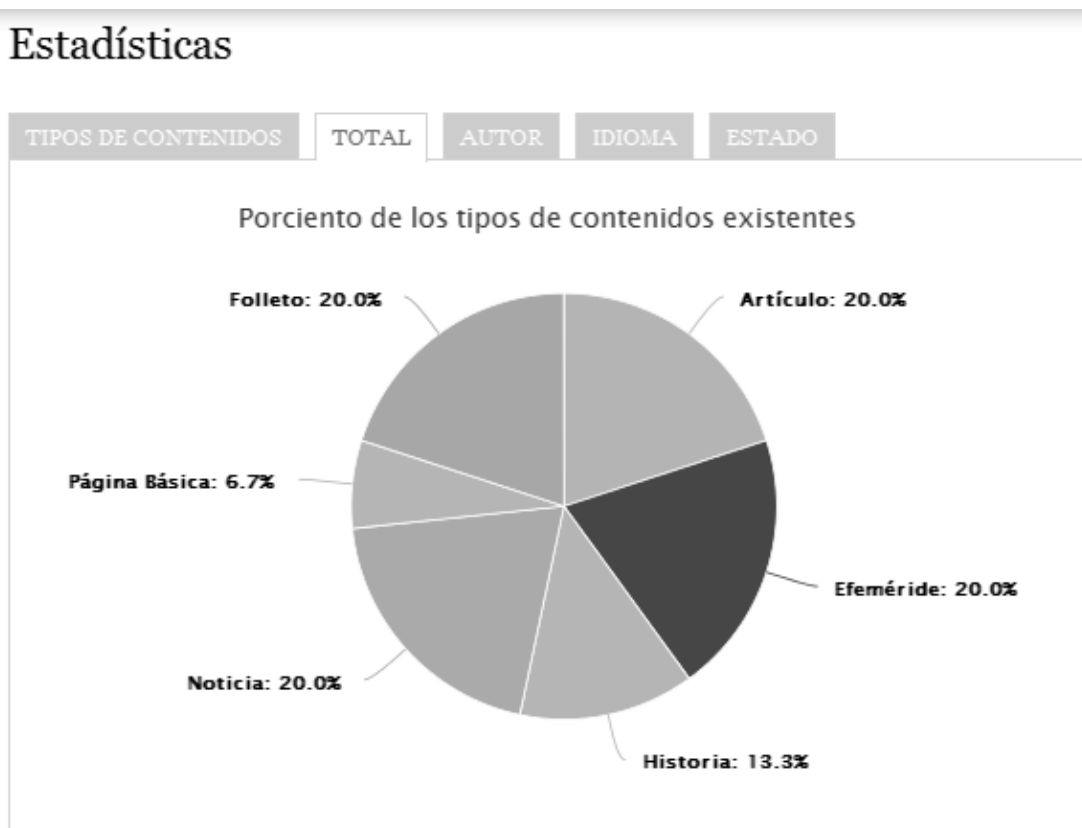


Figura 6: Total de contenidos creados por tipo.

Contenidos del autor

Se muestra un gráfico de pastel que muestra la cantidad de contenidos creados por el usuario autenticado.



Figura 7: Contenidos creados por tipo del usuario autenticado.

3.3 Validación de la propuesta de solución

Los procesos de pruebas de software se destacan como el proceso de establecer datos de entrada al producto de software e inspeccionar los resultados obtenidos, permitiendo determinar si el software funciona realmente como se espera. Las pruebas de software responden fundamentalmente a dos interrogantes, ¿se ha obtenido un buen producto?, ¿se ha desarrollado de forma correcta? Este concepto da lugar al proceso de verificación y validación del software (Zamora, 2011).

Pruebas de integración

Las pruebas de integración son una técnica que permiten probar en conjunto distintos de subsistemas funcionales o componentes del proyecto para verificar que interactúan de manera correcta y que se ajustan a los requisitos especificados (sean estos funcionales o no) (Echavarria, 2014).

El módulo de administración de contenidos en aplicaciones web desarrolladas con Drupal 7 se integra a los sitios web desarrollados con Drupal 7 implementados en cualquier gestor de base de datos que permita el CMS el cual puede estar instalado sobre los sistemas operativos Linux o Windows. A continuación se muestra un ejemplo de las pruebas de integración realizadas.

Tabla 5: Prueba de integración con el CMS Drupal usando el gestor de base de datos MySQL.

Caso de Prueba: Integración con el CMS Drupal usando el gestor de base de datos MySQL.
Sistema/componente al que se integra: CMS Drupal
Condiciones de ejecución: Se debe tener instalado el gestor datos de base de datos MySQL y debe existir conexión con la misma.
Descripción de la prueba: Comprobar que el módulo cree en la base de datos del sitio la tabla correspondiente del módulo (user_preference) y el módulo se ejecute correctamente.
Pasos de ejecución: El módulo se instala /Módulos/manage_content /.
Resultados esperados: Se debe instalar el módulo en el CMS Drupal usando el gestor de base de datos MySQL correctamente sin ningún error.
Evaluación: Prueba satisfactoria

Pruebas de funcionalidad

Las pruebas funcionales aplicadas a la propuesta de solución fueron basadas en las historias de usuarios, donde se recogen los escenarios correspondientes a cada requisito funcional. En correspondencia la metodología AUP-UCI, para la realización de las pruebas se empleó la plantilla de diseño de casos de pruebas, en la que se describe el comportamiento de la aplicación ante determinado flujo de datos.

Casos de prueba

Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Caso de Prueba 1: SC RF1_Filtrar contenido por título.

Descripción general: El sistema permite filtrar los contenidos por título.

Condiciones de ejecución: El sistema debe estar disponible.

Tabla 6: Caso de prueba 1: filtrar contenidos por título.

Escenario	Descripción	Título	Respuesta del sistema	Flujo central
EC 1.1 Filtrar contenidos por título existente.	El sistema filtra contenidos por título de forma correcta.	V Título existente	El sistema filtra los contenidos y le muestra al usuario el contenido con el título correspondiente.	El usuario introduce el título del contenido a filtrar y oprime el botón "Aplicar".
EC 1.2 Filtrar contenidos por título no existente.	El sistema no filtra contenidos por título de forma incorrecta.	I Título no existente.	El sistema no muestra ninguna lista de contenidos.	El usuario introduce el título del contenido a filtrar y oprime el botón "Aplicar".
EC 1.3 Filtrar contenidos por títulos dejando el campo vacío.	El sistema filtra contenidos por título dejando el campo vacíos.	NA	El sistema le muestra al usuario todos los contenidos existentes.	El usuario no introduce el título del contenido a filtrar y oprime el botón "Aplicar".

Tabla 7: Descripción de las variables del caso de prueba 1.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Título	campo de texto	Si	Permite todos los caracteres

Caso de Prueba 7: SC RF7_Definir cantidad de elementos a mostrar.

Descripción general: El sistema permite definir cantidad de elementos a mostrar.

Condiciones de ejecución: El sistema debe estar disponible y el usuario debe estar autenticado en el sistema.

Tabla 8: Caso de prueba 7: definir cantidad de elementos a mostrar.

Escenario	Descripción	Elementos	Respuesta del sistema	Flujo central
EC 7.1 Definir 5 como la cantidad de elementos a mostrar.	El módulo muestra 5 elementos al realizar el filtrado.	V 5	El módulo filtra los contenidos y le muestra al usuario un listado de contenidos que no superan los 5 elementos por página.	El usuario selecciona la cantidad de contenido a mostrar y oprime el botón "Aplicar".
EC 7.2 Definir 10 como la cantidad de elementos a mostrar.	El módulo muestra 10 elementos al realizar el filtrado.	V 10	El módulo filtra los contenidos y le muestra al usuario un listado de contenidos que no superan los 10 elementos por página.	
EC 7.3 Definir 20 como la cantidad de elementos a mostrar.	El módulo muestra 20 elementos al realizar el filtrado.	V 20	El módulo filtra los contenidos y le muestra al usuario un listado de contenidos que no superan los 20 elementos por página.	

EC	7.4	El módulo muestra 40 elementos al realizar el filtrado.	V	El módulo filtra los contenidos y le muestra al usuario un listado de contenidos que no superan los 40 elementos por página.	
Definir	40		40		
como la cantidad de elementos a mostrar.					
EC	7.5	El módulo muestra 60 elementos al realizar el filtrado.	V	El módulo filtra los contenidos y le muestra al usuario un listado de contenidos que no superan los 60 elementos por página.	
Definir	60		60		
como la cantidad de elementos a mostrar.					

Tabla 9: Descripción de las variables del caso de prueba 7.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Elementos	Campo de selección múltiple	no	Puede tomarlos valores 5, 10, 20, 40 y 60.

Caso de Prueba 14: SC RF14_Enviar notificaciones de usuario cuando un contenido ha sido modificado.

Descripción general: El sistema permite enviar un correo de notificación a un usuario cuando se modifica un contenido de su autoría.

Condiciones de ejecución; El sistema debe estar disponible y el usuario debe estar autenticado en el sistema.

Tabla 10: Caso de prueba 14: mostrar notificaciones de usuario cuando un contenido ha sido modificado.

Escenario	Descripción	Notificar	Respuesta del sistema	Flujo central
EC 14.1 Enviar notificación de forma correcta	El módulo envía la notificación de forma correcta.	V si	El sistema edita el contenido y muestra el siguiente mensaje: "Mensaje enviado "	Al finalizar la edición del contenido el usuario oprime el botón "Guardar".
EC 14.2 No poder enviar la notificación.	El módulo no puede enviar la notificación.	V si	El sistema edita el contenido y muestra el siguiente mensaje: "No se pudo enviar el mensaje. Si el problema persiste contactar con el administrador del sitio "	
EC 14.2 No enviar notificación si el autor no la requiere.	El usuario no requiere que lo notifiquen.	NA	El sistema edita el contenido y no muestra ningún mensaje de confirmación de envío.	

Tabla 11: Descripción de las variables del caso de prueba 14.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Notificar	Campo de selección	si	Este campo puede estar activado o desactivado.

Resultado de las pruebas de funcionalidad

En la primera iteración se detectan 16 no conformidades, de ellas 4 de validación, 5 son de error de concordancia, 4 de ortografía y 3 de recomendación. En la segunda iteración se encuentran un total de 8 no conformidades 3 de validación, 2 de error de concordancia, 1 de ortografía y 2 de recomendación. Se

realizó una tercera iteración en la que no se encontraron no conformidades. Como resultado final todas las no conformidades encontradas fueron resueltas, lo que demuestra que el software funciona correctamente.

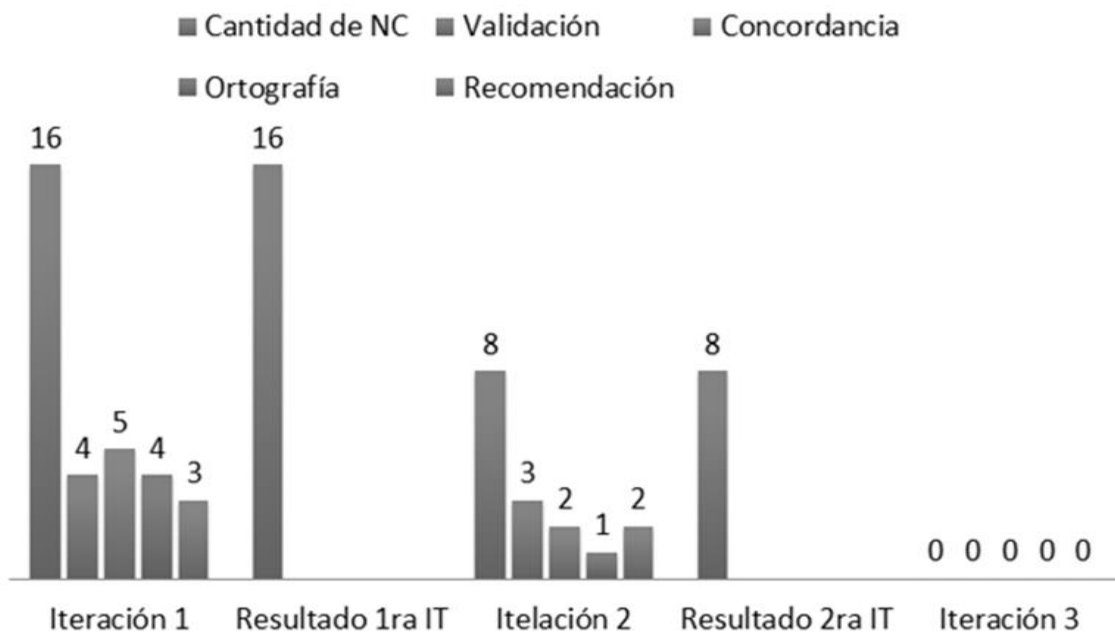


Figura 8: Resultado de las pruebas de funcionalidad.

Prueba de seguridad

Las pruebas de seguridad son un proceso que permite verificar que un sistema de información protege sus datos y funciona de acuerdo a los propósitos para los que fue diseñado. Se encarga de comprobar los siguientes aspectos:

- Nivel de seguridad de la aplicación: Verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido.
- Seguridad del sistema, incluyendo acceso a datos o funciones de negocios y accesos remotos.
- Funciones / Seguridad de Datos: Identificar cada tipo de usuario y las funciones y datos a los que se debe autorizar.

Al ser difícil conocer vulnerabilidades antes de que sucedan existen aplicaciones que ayudan a prevenirlas comprobando los fallos de seguridad más habituales, por lo que para identificar dichas vulnerabilidades se utilizará Acunetix web vulnerability scanner.

Resultados de las pruebas de seguridad

La herramienta mostró resultados dentro de lo que cabe señalar que las amenazas que se detectaron fueron de las credenciales de usuario que se transmiten en un texto plano sin cifrar, además de que la página de inicio de sesión no tiene ningún tipo de protección contra los ataques de contraseña adivinanzas por lo que se recomienda aplicar algún tipo de bloqueo de cuentas después de varios intentos de contraseñas incorrectas. Se considera no realizar otra iteración, debido a que el módulo en sí no presenta ninguna amenaza, sino que depende de las características del sistema donde está instalado.

3.4 Conclusiones

El uso de los estándares de codificación para la implementación del módulo posibilitó adoptar una estructura que contribuye a la calidad del módulo y obtener un código legible y fácil de comprender, con una menor cantidad de errores.

Las pruebas al software permitieron detectar tempranamente las deficiencias del desarrollo, corregir errores y fallos en el funcionamiento del módulo, así como validar su correcto funcionamiento, para asegurar que la propuesta solución sea estable y segura.

Conclusiones Generales

La investigación desarrollada y los resultados obtenidos permiten arribar a las siguientes conclusiones:

- El estudio de los principales aspectos relacionados con los sistemas de gestión de información permitió sentar las bases para el posterior desarrollo del módulo.
- A partir del análisis y diseño del módulo a implementar se obtuvieron los artefactos necesarios para guiar el proceso de desarrollo del mismo.
- Como resultado de la implementación se obtuvo el módulo para la administración de contenidos desarrollado con Drupal 7, que cumple con los 15 requerimientos funcionales identificados en la fase de ejecución.
- La solución fue validada a partir de la ejecución de pruebas de integración, funcionalidad y seguridad las cuales permitieron comprobar el correcto funcionamiento del módulo desarrolladas con Drupal 7 lo que contribuyó a perfeccionar la administración de contenidos en aplicaciones web desarrolladas con el CMS Drupal.

Recomendaciones

Una vez concluida la investigación y el desarrollo de la propuesta de solución, la autora del presente trabajo recomienda:

- Migrar el módulo a versiones superiores de Drupal 7.
- Adicionar las funcionalidades necesarias para poder exportar las estadísticas a PDF.

Bibliografía

- 1and1. 2016.** 1and1.es. [En línea] 26 de mayo de 2016. <https://www.1and1.es/digitalguide/hosting/cms/cms-en-comparativa-los-gestores-de-contenido-mas-usados/>.
- Alvarez, Miguel Angel. 2001.** desarrolloweb.com. [En línea] 9 de mayo de 2001. <http://www.desarrolloweb.com/articulos/392.php>.
- Desarrollo Web. 2001.** desarrolloweb.com. [En línea] 1 de 1 de 2001. <http://www.desarrolloweb.com/articulos/que-es-html.html>.
- Andrés, María Marqués. 2005.** [En línea] 2005. <http://www3.uji.es/~mmarques/f47/apun/node32.html>.
- Barberá, Joaquín. 2014.** diseñowebmurcia1.es. [En línea] 2 de septiembre de 2014. <http://www.diseñowebmurcia1.es/clasificacion-caracteristicas-cms-gestores-de-contenidos/>.
- Butcher. 2008.** M. *Learning Drupal 6 Module Development*. 2008.
- Canez, Nataly. 2016.** drupalitics.blogspot.com. [En línea] 4 de mayo de 2016. <http://drupalitics.blogspot.com/2016/05/drupal-drupal-es-un-sistema-de-gestion.html>.
- Carcasés, Arianne Ferrer. 2016.** Guía para los procesos de Usabilidad que se. 2016.
- Chaffer, J.. 2016.** drupal.org. [En línea] 2016. <http://www.drupal.org/node/547518>.
- Cuerda, Xavier García. 2014.** Mosaic. [En línea] 29 de 11 de 2014. <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>.
- Mosaic. 2004.** mosaic.uoc.edu. [En línea] 29 de noviembre de 2004. <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>.
- Díaz, Orestes Febles. 2013.** es.slideshare.net. [En línea] 31 de julio de 2013. <http://es.slideshare.net/FransGirn/visual-paradign-24825147>.
- Drupal. 2017.** Drupal. [En línea] 2017. <https://www.drupal.org/docs/7/understanding-drupal/overview>.
- Drupal.** drupal.org. [En línea] [Citado el: 6 de 12 de 2016.] <https://www.drupal.org/drupal-7.0/es>.
- E Camacho, C Fabio, N Gabriel. 2014.** Arquitecturas de Software. Guía de estudio. [En línea] 2014. <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>.
- Echavarría, Raul Antonio Ramirez. 2014.** prezi. [En línea] 18 de 5 de 2014. <https://prezi.com/0mpgx-lmytat/pruebas-de-integracion/>.
- Elvira. 2011.** [En línea] 2011. <http://recursostic.educacion.es/observatorio/web/es/software/software-general/1022>.

Erazo, Cristian Fernando Rodríguez. 2012. *DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL PORTAL WEB DE LA EMPRESA DE AUTO VENTAS "AUTO FÁCIL", APLICANDO LA HERRAMIENTA DE DESARROLLO WEB OPEN SOURCE DRUPAL.* Ibarra – Ecuador : s.n., 2012.

Galán, Hector García. 2016. Blog de Arsys. [En línea] 27 de 4 de 2016. <https://blog.arsys.es/que-es-typo3/>.

Genbetadev. genbetadev. [En línea]

Giner, Clara y Elena, Fernández. 1999. *Plataforma Digital De Revistas Científicas.* 1999. Vols. Elena, Fernández.

Guzzo, Fabricio. 2016. <http://www.fabricioguzzo.com/>. [En línea] 2016. <http://www.fabricioguzzo.com/CMS/Drupal>.

Hume, Emilio. 2015. Time Line Web Technology. [En línea] 8 de 2 de 2015.
<http://timelinewebtechnology.blogspot.com/2015/02/surgimiento-del-internet.html>.

Ingeniería de Software. 2016. ingenieriadesoftware.mex.tl. [En línea] 2016. [Citado el: 6 de 12 de 2016.]
http://ingenieriadesoftware.mex.tl/63758_AUP.html.

Jacobson, Ivar. 2013. *El proceso unificado de desarrollo de software.* 2013. Vol. 3.

Joomla! Documentation. 2016. Joomla! Documentation. [En línea] 7 de 4 de 2016.
https://docs.joomla.org/J3.x:Joomla_Core_Features/es.

kyve. 2014. drupal.org. [En línea] 18 de 3 de 2014. <http://drupal.org/easychart/>.

Larman, C. 2000. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 2000.

Craig Larman. 2016. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 2016.

Lazaro, Juliana Monteiro. 2001. desarrolloweb.com. [En línea] 1 de 1 de 2001.
<http://www.desarrolloweb.com/articulos/26.php>.

Martinez, Rafael. 2010. postgresql.org.es. [En línea] 2 de octubre de 2010.
http://www.postgresql.org.es/sobre_postgresql.

mstef. 2010. drupal.org. [En línea] 6 de 8 de 2010. <http://drupal.org/quant/>.

Olzinelles. factoressencial.com. [En línea]

Opensuse. 2011. Software Opensuse. [En línea] 2011. <https://es.opensuse.org/Apache>.

Peréz, Javier. 2008. [En línea] 20 de febrero de 2008.

quicksketch. 2008. drupal.org. [En línea] 11 de 3 de 2008. <http://drupal.org/charts/>.

Reingart, Mariano. 2016. postgresql.org. [En línea] 6 de 12 de 2016.
www.postgresql.org.ar/trac/wiki/PgAdmin?format=pdfarticle.

Reyero, José. reyero.net. [En línea]

Rodríguez, Fran Gil. 2011. *Experto en Drupal 7 Nivel Inicial*. 2011.

Scholarium. 2017. Scholarium. [En línea] 2017. <http://scholarium.info/wordpress-sistema-de-gestion-de-contenidos-cms/>.

scottrigby. 2010. drupal.org. [En línea] 27 de 9 de 2010. <http://drupal.org/charts/>.

Seaq. Seaq. [En línea] [Citado el: 2016 de 12 de 6.] <http://www.seaq.co/acunetix.html>.

Trazada. 2013. Trazada. [En línea] 23 de 4 de 2013. <http://trazada.com/ventajas-de-utilizar-un-cms/>.

web, Desarrollo. Desarrollo web. [En línea]

Zamora, J. 2011. Análisis de los procesos de verificación y validación en las organizaciones. [En línea] 2011. <http://orff.uc3m.es/handle/10016/12880>.

Zamora., J. 2011. Análisis de los procesos de verificación y validación en las organizaciones. [En línea] 2011. <http://orff.uc3m.es/handle/10016/12880>.

Glosario de términos

A

API: Interfaz para programación de aplicaciones (*Application Programming Interface*). Es un conjunto de funciones y procedimientos que ofrece determinada biblioteca para ser utilizada por otro software.

AUP: Metodología de desarrollo el cual es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

AUP_UCI: Es una versión de la metodología de desarrollo ágil de software AUP, que se adapta al ciclo de vida definido para la actividad productiva de la UCI.

B

BD: Base de Datos.

C

CMS: Sistema Gestor de Contenidos. CMS es un software para el manejo de contenidos de sitios web directamente desde el servidor. Sistema Gestor de Contenidos. CMS es un software para el manejo de contenidos de sitios web directamente desde el servidor.

CASE: Ingeniería de Software Asistida por Ordenador (*Computer Aided Software Engineering*), aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

H

HTTP: Protocolo de transferencia de hipertexto (*Hypertext Transfer Protocol*). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la *World Wide Web*.

HTML: Lenguaje de marcas hipertextuales (*Hypertext Markup Language*), diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Herramientas: Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

I

IDE: Entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

Internet: Conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP.

J

JavaScript: Lenguaje interpretado orientado a las páginas web, para realizar tareas y operaciones en el marco de la aplicación cliente.

M

Módulo: Parte de un sistema que manipula ciertos procesos relacionadas entre sí, con el objetivo de lograr un mejor control y una mejor organización de estos.

Modelo: Arquetipo o punto de referencia para imitarlo o reproducirlo.

P

Proceso: Secuencia de actividades que tienen un marcado inicio y fin.

PHP: lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

R

Requisitos: Capacidades, condiciones o cualidades que el sistema debe cumplir y tener.

RSS: es un formato para la sindicación de contenidos de páginas web. Sus siglas responden a *Really Simple Syndication*.

RUP: Proceso Unificado Racional (*Rational Unified Process*) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML.

S

SGBD: Sistema Gestor de Bases de Datos.

U

UML: Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

URL: Localizador uniforme de recurso (*Uniform Resource Locator*). Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

W

Web 2.0: es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones web enfocadas al usuario final.

X

XML: Lenguaje de marcas extensible (*Extensible Markup Language*), es un metalenguaje extensible de etiquetas, una manera de definir lenguajes para diferentes necesidades.

Anexos

Anexo 1: Historia de Usuario: Filtrar contenidos por tipo.

Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Filtrar contenidos por tipo.
Modificación de Historia de Usuario Número: 1	
Usuario: Anabel García García	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 días.
Riesgo en Desarrollo: Alto	Puntos Reales: 2 días.
Descripción: La HU inicia cuando el usuario después de autenticado desea buscar un contenido específico por el rasgo tipo. El usuario accede al bloque de administración, ubicado en la parte superior del portal, selecciona la opción: Contenidos. El sistema le muestra una interfaz con los criterios disponibles para filtrar el contenido. El usuario selecciona el campo de texto con el parámetro Tipo y una vez seleccionado el tipo en la lista que se despliega y haber seleccionado el botón: Aplicar, el sistema le mostrará una lista con los contenidos que sean de ese tipo, y entonces habrá culminado la HU.	
Observaciones: 1- El usuario no introduce ningún tipo y el sistema le muestra toso los contenidos existentes.	
Interfaz:	

Panel de control **Content** Estructura Apariencia Usuarios Módulos Configuración Informes Ayuda Cerrar sesión
Bienvenido, root

Lista de contenidos CONTENIDO COMENTARIOS

➕ Agregar contenido

Título

Tipo

Publicado

Idioma

Autor

Separar por coma la lista de usuarios

Creación a partir del

Ejemplo: 06/20/2017

hasta el

Ejemplo: 06/20/2017

Elementos

OPERACIONES

Seleccione algún filtro y haga click en Filtrar para ver los resultados

Anexo 2: Historia de Usuario: Filtrar contenidos por autor.

Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Filtrar contenidos por autor.
Modificación de Historia de Usuario Número: 1	
Usuario: Anabel García García	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 días.
Riesgo en Desarrollo: Alto	Puntos Reales: 2 días.

Descripción: La HU inicia cuando el usuario después de autenticado desea buscar un contenido específico por el rasgo autor. El usuario accede al bloque de administración, ubicado en la parte superior del portal, selecciona la opción: Contenidos.

El sistema le muestra una interfaz con los criterios disponibles para filtrar el contenido. El usuario selecciona el campo de texto con el parámetro Autor y una vez introducido el autor que desea buscar y haber seleccionado el botón: Aplicar, el sistema le mostrará una lista con los contenidos que hayan sido escrito por ese autor, y entonces habrá culminado la HU.

Observaciones:

1- El usuario introduce el nombre de un autor que no existe y el sistema no le muestra ningún resultado

Interfaz:

Panel de control **Content** Estructura Apariencia Usuarios Módulos Configuración Informes Ayuda Cerrar sesión
Bienvenido, root

Lista de contenidos **CONTENIDO** COMENTARIOS

+ Agregar contenido

Título

Tipo - Seleccione -
Publicado - Seleccione -
Idioma - Seleccione -

Autor

Separar por coma la lista de usuarios

Creación a partir del

Ejemplo: 06/20/2017

hasta el

Ejemplo: 06/20/2017

Elementos 10 **Filtrar**

OPERACIONES
- Seleccione una operación - **Ejecutar**

Seleccione algún filtro y haga click en Filtrar para ver los resultados

Anexo 3: Historia de Usuario: Mostrar notificaciones de usuario cuando un contenido ha sido eliminado.

Historia de Usuario	
Número: HU_15	Nombre Historia de Usuario: Mostrar notificaciones de usuario cuando un contenido ha sido eliminado.
Modificación de Historia de Usuario Número: 1	
Usuario: Anabel García García	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 días.
Riesgo en Desarrollo: Alto	Puntos Reales: 2 días.
<p>Descripción: La HU inicia cuando el usuario después de autenticado desea eliminar algún contenido. En la página principal del sistema después de identificar el contenido que quiere eliminar selecciona la opción Eliminar el sistema le preguntará si está seguro que desea eliminar el contenido y selecciona el botón aceptar y el sistema le envía una correo de confirmación al Autor del artículo diciendo que este ha sido eliminado entonces habrá culminado la HU.</p>	
<p>Observaciones:</p> <p>1-El usuario elimina un contenido y el sistema muestra un mensaje en la página principal diciendo que el contenido ha sido eliminado.</p> <p>2-El usuario cancela la operación de editar el contenido y no se le envía correo al autor.</p>	
<p>Contenido del Correo:</p> <p>Asunto: Contenido eliminado.</p> <p>Cuerpo: Se eliminó el contenido: (nombre del contenido), por el usuario: (Nombre del usuario que eliminó el contenido).</p>	

Anexo 4: Caso de Prueba: Integración con el CMS Drupal usando el gestor de base de datos PostgreSQL.

Caso de Prueba: Integración con el CMS Drupal usando el gestor de base de datos PostgreSQL.
Sistema/componente al que se integra: CMS Drupal.
Condiciones de ejecución: Se debe tener instalado el gestor de base de datos PostgreSQL y debe existir conexión con la misma.
Descripción de la prueba: Comprobar que el módulo cree en la base de datos del sitio la tabla correspondiente del módulo (user_preference) y el módulo se ejecute correctamente.
Pasos de ejecución: El módulo se instala /Módulos/manage_content /.
Resultados esperados: Se debe instalar el módulo en el CMS Drupal usando el gestor de base de datos PostgreSQL correctamente sin ningún error.
Evaluación: Prueba satisfactoria

Anexo 5: Caso de Prueba: Filtrar contenidos por tipo.

Escenario	Descripción	Tipo de contenido	Respuesta del sistema	Flujo central
EC 3.1 Filtrar contenidos por tipo de contenido existente.	El módulo filtra contenidos por tipo de forma correcta.	V	El módulo filtra los contenidos y le muestra al usuario el contenido con el tipo de contenido correspondiente.	El usuario introduce el tipo del contenido a filtrar y oprime el botón "Aplicar".
		Tipo de contenido existente		
EC 3.2 Filtrar contenidos por tipo de contenido no existente.	El módulo no filtra contenidos por tipo de contenido de forma incorrecta.	I	El módulo no muestra ninguna lista de contenidos.	El usuario introduce el tipo de contenido a filtrar y oprime el botón "Aplicar".
		Tipo de contenido no existente.		
EC 3.3 Filtrar contenidos por tipo de contenidos dejando el campo vacío.	El módulo filtra contenidos por tipo de contenido dejando el campo vacíos.	NA	El módulo le muestra al usuario todos los contenidos existentes.	El usuario no introduce el tipo de contenido a filtrar y oprime el botón "Aplicar".

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Tipo de contenido	campo de texto	Si	Permite todos los caracteres

Anexo 6: Caso de Prueba: Filtrar contenidos por autor.

Escenario	Descripción	Autor	Respuesta del sistema	Flujo central
EC 4.1 Filtrar contenidos por autor existente.	El módulo filtra contenidos por autor de forma correcta.	V	El módulo filtra los contenidos y le muestra al usuario el contenido del autor correspondiente.	El usuario introduce el autor del contenido a filtrar y oprime el botón "Aplicar".
		Autor existente		
EC 4.2 Filtrar contenidos por autor no existente.	El módulo no filtra contenidos por autor de forma incorrecta.	I	El módulo no muestra ninguna lista de contenidos.	El usuario introduce el autor del contenido a filtrar y oprime el botón "Aplicar".
		Autor no existente.		
EC 4.3 Filtrar contenidos por autores dejando el campo vacío.	El módulo filtra contenidos por autor dejando el campo vacíos.	NA	El módulo le muestra al usuario todos los contenidos existentes.	El usuario no introduce el autor del contenido a filtrar y oprime el botón "Aplicar".

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Autor	campo de texto	Si	Permite todos los caracteres

Anexo 7: Caso de Prueba: Filtrar contenidos por estado.

Escenario	Descripción	Estado	Respuesta del sistema	Flujo central
EC 5.1 Filtrar contenidos por estado existente.	El módulo filtra contenidos por su estado (publicado o no publicado) de forma correcta.	V	El módulo filtra los contenidos y le muestra al usuario el contenido con el estado correspondiente.	El usuario selecciona el estado del contenido a filtrar y oprime el botón "Aplicar".
		Estado existente		
EC 5.2 Filtrar contenidos por estados dejando el campo vacío.	El módulo filtrar contenidos por estado dejando el campo vacíos.	NA	El módulo le muestra al usuario todos los contenidos existentes.	El usuario no selecciona el estado del contenido a filtrar y oprime el botón "Aplicar".

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Estado	Campo de selección	Si	El campo puede estar vacío o presentar uno de los estados disponibles.

Anexo 8: Interfaz: tipos de contenidos.

Estadísticas



Anexo 9: Interfaz: porcentaje de contenidos por idioma.

Estadísticas

TIPOS DE CONTENIDOS

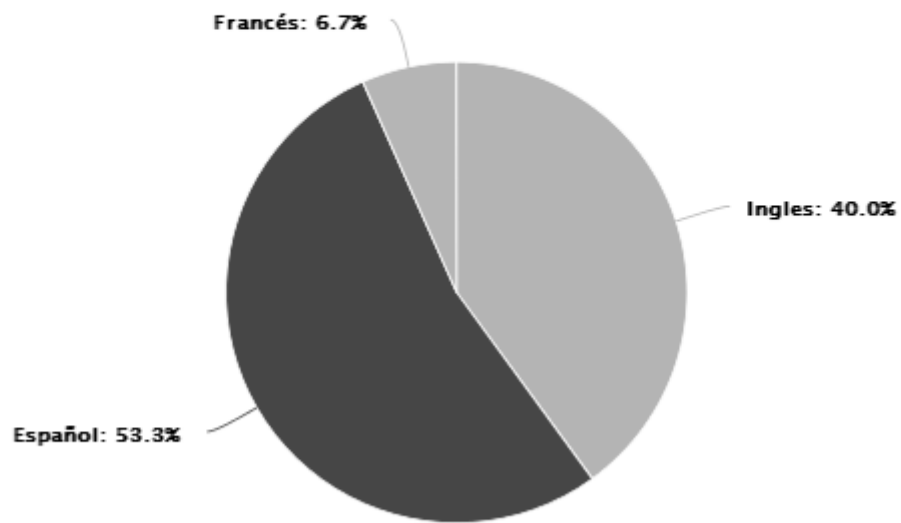
TOTAL

AUTOR

IDIOMA

ESTADO

Porcentaje de contenidos por idiomas



Anexo 10: Interfaz: porcentaje de contenidos por estado.

Estadísticas

