

Universidad de las Ciencias Informáticas Facultad 1



Título: “Plataforma de Publicación Web para el periódico Granma”

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

Autor(es):

Rosana Mir Chávez

Ricardo Valdivia Freyre

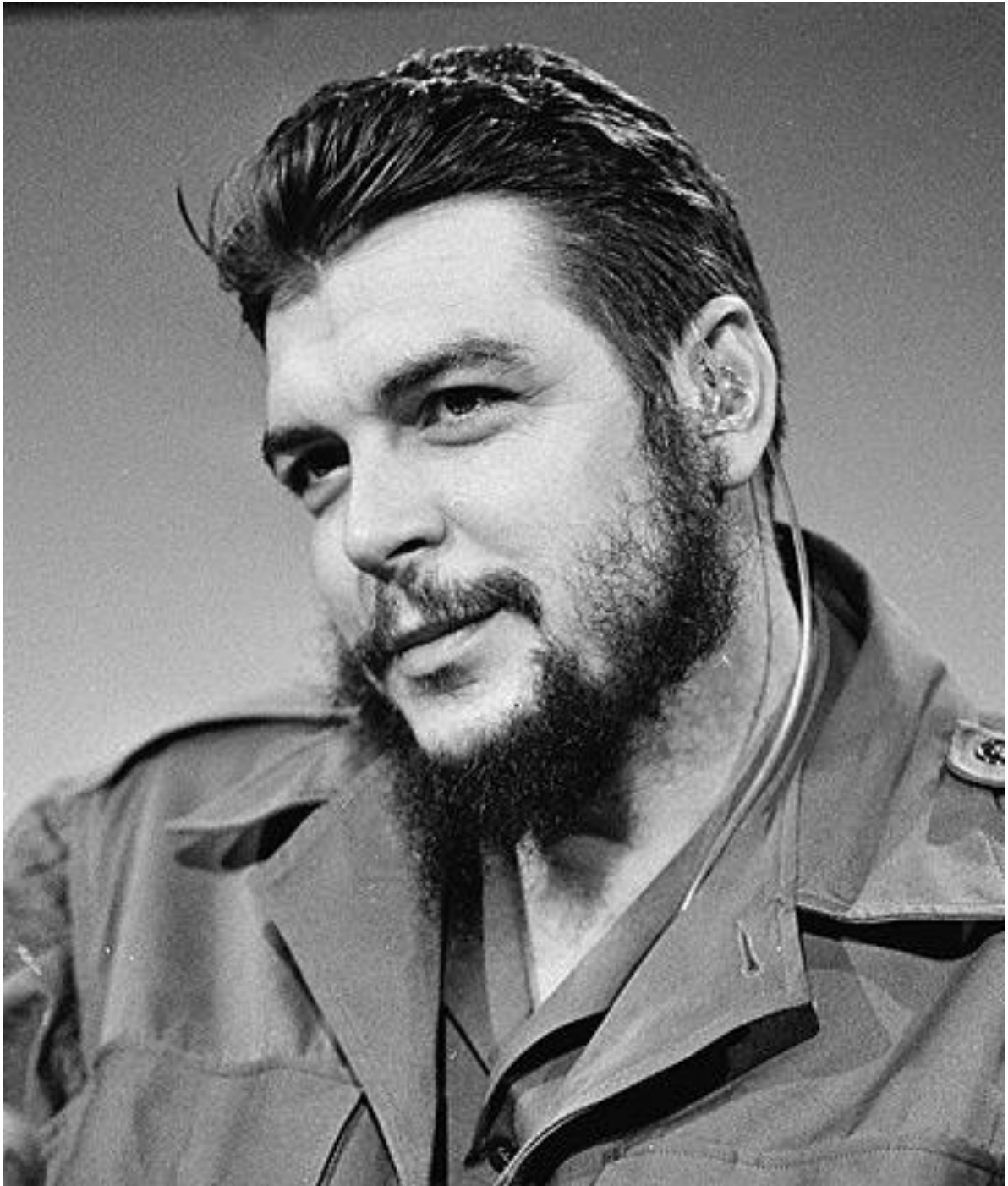
Tutor(es):

Ing. Abel Tellechea Palenzuela

Ing. Daneysi Granado Rodríguez

Ciudad de La Habana, Junio 2012

“Año 54 de la Revolución”



"Nuestra libertad y su sostén cotidiano tienen color de sangre y están henchidos de sacrificio."

Ernesto Ché Guevara

➤ **Abel Tellechea Palenzuela**

Graduado en Ingeniería en Ciencias Informáticas en el año 2010, recién graduado en adiestramiento de la UCI. En la esfera docente ha impartido cursos del CMS Drupal, en la productiva ha participado en el desarrollo de varios proyectos donde ha ejercido como líder de proyecto, programador y analista indistintamente. Ha sido oponente de tres trabajos de diploma.

Correo electrónico: atellechea@uci.cu

➤ **Daneisy Granado Rodríguez**

Graduado de Ingeniero en Ciencias Informáticas en el año 2007, profesora instructora del Centro de Ideoinformática de la Facultad 1. En la esfera docente ha impartido asignaturas de la disciplina de Ingeniería y Gestión de Software. En el área de investigación – producción ha trabajado los temas relacionados con el Periodismo Digital, el Posicionamiento y la Analítica web; ha tutorado trabajos de diploma dentro de este campo, ha ejercido también como miembro de tribunal y oponente de varias tesis de grado.

Correo electrónico: dgranado@uci.cu

Agradecimientos

Rosana

A Dios, que me ha guiado siempre por el camino correcto.

A mi madre y mi padre, por tanto apoyo cuando más lo necesité y porque me han enseñado a tener paciencia cuando se quiere alcanzar algo.

A mis hermanitas lindas por dedicarme su tiempo y cariño.

A mi nene, por aceptarme con mis virtudes y defectos los últimos 3 años, gracias por tanto amor y comprensión.

A toda mi familia, por apoyarme y preocuparse siempre por mí.

A mi suegri y mi cuñi, por ser tan buenas personas y ayudarme tanto y a toda la familia de mi novio por el cariño que he recibido de ellos.

A Sandry, Yoanni y Yasniel por brindarme su gran amistad y a todos los amigos que me apoyaron en estos 5 años.

A los profesores Luis Domínguez, Yankiel Aparicio, Adrián Garbey y Gilberto Lissabet, por su ayuda en los momentos más difíciles.

A mi tutora y tribunal por guiarme durante el proceso de desarrollo de la tesis en la redacción y presentación de la misma.

A todos los que una forma u otra contribuyeron en mi formación.

Ricardo

Mencionar nombres es algo injusto cuando existen tantas personas que me han ayudado a llegar hasta aquí y de modo sencillo pero sincero a todos por igual, “Gracias por su apoyo”.

Dedicatoria

Rosana

A mi mamita por ser mi fuente de inspiración por la cual me guío para luchar en la vida y a mi papito por todo el apoyo que me ha dado para seguir sus pasos y formarme como ingeniera.

Ricardo

A todos los que de un modo u otro me han brindado su apoyo.

Declaración de Autoría

Declaro ser autor del presente trabajo de diploma y autorizo al centro CIDI de la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Rosana Mir Chávez

Firma del Autor

Ricardo Valdivia Freyre

Firma del Autor

Ing. Daneysi Granado Rodríguez

Firma del tutor

Ing. Abel Tellechea Palenzuela

Firma del Cotutor

La prensa digital en la actualidad ha marcado la diferencia en las variadas formas de transmitir informaciones y noticias. En Cuba se ha sabido aprovechar su importancia y varios medios de prensa poseen sitios web en Internet como el periódico Granma, el cual presenta algunas deficiencias que hacen engorroso el trabajo en la aplicación al momento de insertarle contenidos y que los usuarios naveguen por el sitio.

En la Universidad de las Ciencias Informáticas se está desarrollando un sistema digital para la gestión de todo el proceso editorial del periódico. Está compuesto por un Sistema de Gestión Editorial, un Sistema de Gestión Documental y una Plataforma de Publicación Web.

En el presente trabajo se tiene como objetivo la creación de la plataforma para publicar en Internet los contenidos gestionados en el Sistema de Gestión Editorial, desarrollado sobre el CMS Drupal y utilizando PostgreSQL como sistema gestor de bases de datos. Como lenguajes de programación se emplean PHP5 en el lado del servidor y Javascript en el cliente, además de AJAX, CCS2 y XHTML para gestionar la presentación visual de ambos lados y se logra la integración con el Sistema de Gestión Editorial mediante un servicio web para la inserción o modificación de los contenidos.

Palabras clave

Granma, Plataforma de Publicación Web, Prensa digital.

Introducción	1
Capítulo 1: Fundamentación teórica.....	6
1.1 Definiciones de interés	6
1.2 Tecnologías, herramientas y lenguajes de programación a investigar	8
1.2.1 Sistemas de Gestión de Contenidos (CMS).....	8
1.2.2 Frameworks (CMF).....	10
1.2.3 Gestores de base de datos	11
1.2.4 Servidor Web.....	12
1.2.5 Administración de base de datos	13
1.2.6 Lenguajes principales para el desarrollo	14
1.2.7 Entorno de Desarrollo Integrado (IDE).....	17
1.2.8 Servicio Web	17
1.2.9 Herramientas de desarrollo.....	18
1.2.10 Metodología de desarrollo de software.....	19
1.2.11 Herramienta de modelado visual	22
Capítulo 2: Propuesta de solución y desarrollo ágil del sistema.....	25
2.1 Propuesta de solución.....	25
2.2 Proceso de desarrollo del proyecto basado en SXP	26
2.3 Modelo de dominio.....	26
2.4 Levantamiento de Requisitos	27
2.4.1 Requisitos funcionales	28
2.4.2 Requisitos no funcionales	29
2.5 Historias de Usuario	32
2.6 Visión general de la arquitectura	34
2.7 Diseño	37
2.7.1 Diagrama de paquetes.....	37
2.7.2 Diseño de la base de datos.....	44
2.8 Implementación	46
2.8.1 Diagrama de despliegue.....	47
2.8.2 Diagrama de componentes	48
Capítulo 3: Pruebas.....	52
3.1 Prueba de software	52
3.2 Casos de prueba	52
3.3 Pruebas de rendimiento	62
3.3.1 Herramienta.....	62
3.3.2 Tipos de Pruebas.....	62
3.4 Resultados obtenidos	64
Conclusiones	65
Recomendaciones	66

Referencias Bibliográficas	67
Bibliografía	69
Glosario de términos	71
Anexo.....	73
Anexo 1. Lista de Reserva del Producto.....	73
Anexo 2. Historias de usuario.....	76
Anexo 3. Lista de Riesgos.....	76
Anexo 4. Diagramas de clases del diseño con estereotipos web.....	79
Anexo 5. Paquetes que contienen las clases persistentes.....	90
Anexo 6. Descripción de las tablas.....	92
Anexo 7. Diagramas de componente.	98
Anexo 8. Casos de prueba de funcionalidad.....	100
Anexo 9. Casos de prueba de integración.....	100

La prensa es un medio que desde sus comienzos ganó buena aceptación entre sus consumidores, esto hizo que fuera evolucionando y mejorando a la par de las nuevas tecnologías y de este modo fueran surgiendo diversas modalidades, tales como la impresa, la radial o la televisiva. Todas con el objetivo de transmitir información o noticias.

En sus comienzos el tema de la distribución y la extensión de los contenidos publicados fueron algunas de las principales limitantes para la prensa impresa. En el pasado, quienes tenían acceso a la prensa la recibían con gran atraso si no vivían cerca de donde se imprimían los ejemplares, ya sea por la distancia que debía ser transportada o por las limitaciones de los medios de transporte de la época. Por motivos del tamaño de la impresión muchas veces la información no era lo suficientemente completa como para llegar al público que la leía, pudiendo causar confusión o también equivocadas conclusiones.

La época contemporánea, revolucionada por las tecnologías digitales, trae consigo la aparición de una nueva corriente: “la prensa digital”, caracterizada por la publicación de contenidos en sitios web en Internet; combina de una forma atractiva y novedosa textos, imágenes y vídeos enriqueciendo la información ofrecida y sin límites de tamaño (Rost, 2003).

Internet ofrece una serie de facilidades para transmitir datos en segundos por todo el mundo, lo cual es aprovechado en toda su gama de sentidos por el periodismo, ahora llamado periodismo digital. Este comenzó asimilando las mismas características que traía hasta el momento pero con el paso del tiempo y su desarrollo, adquirió características distintivas. El periodismo digital es más activo, dinámico e hiperactivo. No solo es el autor quien lo desarrolla sino también todos aquellos que interactúan con el contenido mismo (Bonvin, 2007).

La prensa digital tiene una limitante que la afecta y esta es la accesibilidad a las tecnologías para revisar este tipo de formato, aún con la rapidez con que se transmite la información alrededor del mundo. Se puede decir que en estos momentos la distribución por este medio no es un problema grave pues puede ser consultada desde cualquier lugar del mundo con acceso al Internet y los formatos digitales existentes facilitan su descarga para revisiones posteriores o su transportación.

La prensa digital ayuda a la creación de comunidades tanto para fomentar su desarrollo como para el consumo personal en tiempo real e inmediato. Un hecho puede estar ocurriendo en un momento y en cuestión de segundos aparecer en la red con variados detalles y puntos de vista.

Cuba tiene una larga historia en temas de prensa y periodismo, remontándose a su etapa colonial y para la actualidad posee una amplia experiencia en el campo de la prensa digital. Desde el año 1997 se viene promoviendo la digitalización de los medios de prensa nacionales y provinciales y con la realización de talleres en el año 2001 sobre periodismo digital, se promueven paradigmas de diseño y estrategias de posicionamiento para permitir un mayor éxito en la publicación de prensa digital, que ésta sea reconocida, tenga claridad y motive a las personas que navegan por la red a conocer la realidad cubana.

La publicación de noticias en Internet es una competencia constante en la cual muchos medios de prensa del mundo están involucrados, los medios cubanos necesitan aportar primicias y para lograrlo se deben mejorar los recursos existentes y mantenerlos actualizados. Esta competición puede influir mucho en la forma de pensar de las personas, dado el impacto que tienen los contextos en que se definen las noticias y cada versión de una misma historia.

Las imprentas cubanas diaria y semanalmente liberan a las calles del país ejemplares tales como Juventud Rebelde, Trabajadores, Granma y otros, con noticias sobre la actualidad internacional, nacional y regional. La capacidad de actualización en tiempo real que posibilita el Internet permite que nuevas noticias sean publicadas entre ejemplares en las versiones digitales, enriqueciendo con una mayor variedad de contenidos las que provienen de la edición impresa.

El trabajo de los medios de prensa cubanos se basa en un código de ética muy riguroso, tratando siempre de mostrar la verdad a las personas lo más fiel al hecho real, de ahí el prestigio que posee y se debe mantener en la red cada día. Muchos han sido y son los problemas que afronta la prensa en el país como la carencia de recursos o tecnologías modernas pero eso no imposibilita que se realice un buen trabajo, siempre manteniendo una postura fiel a los principios definidos para su desarrollo.

El periódico Granma, Órgano Oficial del Comité Central del Partido Comunista de Cuba (PCC) es el más importante medio de prensa del país y cuenta con una versión digital en Internet cuyo objetivo es divulgar los contenidos de la edición impresa diaria y puede incluir contenidos que no pudieron ser publicados por variadas razones, como espacio en la edición o llegar fuera del tiempo establecido para su análisis. Los contenidos llegan a la Redacción Digital del Periódico Granma ya analizados y aprobados para su posterior publicación, en este proceso participan alrededor de 110 trabajadores.

La Redacción Digital tiene como objetivo publicar estos contenidos en la edición digital. Este proceso se realiza de forma manual en una aplicación que no cuenta con los requerimientos necesarios ni actualizados en correspondencia a las tecnologías digitales existentes, haciéndose lento y complejo el

proceso y en consecuencia se pueden realizar unas pocas actualizaciones en un día. Es una desventaja con respecto a otros medios de prensa que publican noticias constantemente en la web. El resultado de esta herramienta es un sitio web de páginas estáticas y un bajo nivel de organización y clasificación de contenidos, así no son aprovechados los avances tecnológicos actuales para el uso y desarrollo del Internet.

Con el objetivo de solucionar esta situación en la Universidad de las Ciencias Informáticas (UCI) se propuso desarrollar un sistema digital para la redacción del periódico Granma. El sistema se compone a su vez de tres sistemas que abarcan el ciclo completo de la gestión del proceso editorial de la prensa y se complementan unos con otros; estos son: el Sistema de Gestión Editorial, una aplicación informática basada en tecnologías web para la gestión de contenidos referentes al proceso de edición de materiales periodísticos del periódico Granma; el Sistema de Gestión Documental que se encarga de la gestión de imágenes con el objetivo de crear un archivo de fotografías que pueda ser utilizado para apoyar los procesos editoriales del medio y una Plataforma de Publicación Web, donde se administrarían los contenidos que se publican en la edición digital del diario, tomando lo necesario de sus complementarios.

Un portal web destinado a la publicación de contenidos para el periódico fue desarrollado con el CMS WordPress, siendo el resultado de la tesis “Periódico Digital Granma” de los autores Roberto de la Rosa Carbonell y Yasser Antúnez Uffort y continuada por miembros del Departamento de Soluciones Informáticas para Internet (SINI) del centro de Ideoinformática de la UCI. Es una solución temporal en vías de mejorar la estética del portal existente. Para insertar noticias al sitio web se deben copiar las noticias en formato de texto que son generadas por el sistema de gestión editorial y pegarlas textualmente en el portal, dado que no tiene implementada una forma de comunicarse con el Sistema de Gestión Editorial. El sitio web existente dificulta y ralentiza las capacidades de actualización y publicación de noticias en el periódico digital. Además no se existe un modo de comunicarse con el sistema desarrollado para la gestión editorial para la publicación rápida de las noticias.

Atendiendo a la situación problemática antes mencionada, se plantea el siguiente **problema de investigación**, ¿Cómo publicar en Internet los contenidos que se gestionan en la redacción del periódico Granma? Se tiene como **objeto de estudio** el desarrollo de aplicaciones web y el **campo de acción** el desarrollo de aplicaciones web para medios de prensa.

Se plantea como **objetivo general**, desarrollar una Plataforma de Publicación Web que posibilite la publicación en Internet de los contenidos que se gestionan en la redacción del periódico Granma y como **objetivos específicos** se plantearon los siguientes:

- Elaborar una sistematización teórica sobre las características de la prensa digital.
- Analizar las tecnologías factibles para el desarrollo de portales web.
- Efectuar el Análisis y el Diseño a la Plataforma de Publicación Web para el periódico Granma.
- Implementar la Plataforma de Publicación Web para la publicación de contenidos periodísticos.
- Realizar pruebas de funcionalidad, integración y rendimiento a la Plataforma de Publicación Web.

Se espera como **posible resultado** una Plataforma de Publicación Web que logre la publicación de los contenidos que se gestionan en la redacción del periódico Granma, para lograrlo se ha de cumplir con los objetivos trazados anteriormente por lo que se hace necesario desarrollar las siguientes **tareas de investigación**:

1. Análisis de las posibilidades de comunicación que brinde el Sistema de Gestión Editorial del Periódico Granma.
2. Elaboración de sistematización teórica de las tecnologías y herramientas necesarias para el desarrollo de la Plataforma de Publicación Web del periódico Granma para la publicación de contenidos periodísticos.
3. Implementación de las funcionalidades de la Plataforma de Publicación Web.
4. Implementación de la integración de la Plataforma de Publicación Web con el Sistema de Gestión Editorial.
5. Diseño y aplicación de las pruebas de funcionalidad, integración y rendimiento a la Plataforma de Publicación Web.

Se **Justifica la Investigación** partiendo de la premisa que la creación de una Plataforma de Publicación Web para el Periódico Granma, garantizará la publicación en Internet de los contenidos que se gestionan en la redacción del periódico Granma.

Se prevé dar cumplimiento a las tareas anteriormente mencionadas usando los siguientes **métodos teóricos**:

El **Analítico-Sintético** para comprender el funcionamiento de los distintos procesos que se desarrollan en el periódico Granma y generar una conclusión a través de la síntesis de las posibles herramientas que se pueden usar para el desarrollo de la aplicación.

El **Histórico-lógico** para analizar y estudiar los antecedentes y evolución de los elementos asociados al desarrollo e implementación de periódicos digitales.

La **Modelación** para representar los procesos definidos en el sistema, simplificando la realidad y facilitando la comprensión de los mismos.

El contenido de este documento se estructura de la forma siguiente:

Capítulo 1: “Fundamentación teórica”, donde se indagan una serie de conceptos generales relacionados con la prensa digital, se analizan las nuevas tendencias a tener en cuenta para mejorar el desarrollo de los periódicos digitales y se hace un estudio de las diferentes herramientas, tecnologías y metodologías a seleccionar, para dar solución al problema planteado.

Capítulo 2: “Propuesta de solución y desarrollo ágil del sistema”, se identifican, seleccionan y describen los servicios que brindarán el sistema, el modelo de negocio, los requisitos del sistema, la arquitectura del sistema, los patrones arquitectónicos y de diseño, las clases a utilizar en la solución y la relación que existe entre ellas.

Capítulo 3: “Pruebas”, las funcionalidades implementadas son sometidas a pruebas de funcionalidad para su posible validación, con el objetivo de arribar posteriormente a las conclusiones de los resultados arrojados por las comprobaciones realizadas.

Introducción

En el presente capítulo se analizarán las diferentes tecnologías útiles para desarrollar periódicos digitales, posteriormente se seleccionarán las que sean más factibles en la solución del problema. Para elaborar este análisis se tendrán en cuenta las metodologías de desarrollo de *software*, sistemas de bases de datos, lenguajes de programación y tecnologías para el desarrollo web.

1.1 Definiciones de interés

Periódicos digitales

El periódico digital es un medio de comunicación que ha recibido desde su aparición muchas definiciones. Sea periódico digital o sitio de noticias puede resumirse en un medio que renueva sus contenidos y los organiza en períodos de tiempo variados y categorías para su fácil acceso. Su distinción “digital” lo marca como diferente, al incorporar no solo textos sino también sonidos e imágenes en sus contenidos. Como medio de comunicación es diferente de los existentes, si bien hereda de otros medios de prensa los contenidos, formas y estructuras; tiene soporte distinto, permite otras formas de descripción de la actualidad, su proceso de recepción es diferente y requiere un tratamiento periodístico distinto.

Se caracteriza por seis elementos principales: permite la **interactividad** con los contenidos y con otros individuos; organiza su discurso en forma **hipertextual**; ofrece una **actualidad múltiple**, en el sentido que da un menú de contenidos con diferentes temporalidades internas; no tiene límites de tiempo ni de espacio por lo que permite una extensa **documentación** de la actualidad; puede combinar distintos recursos **multimedia** y permite cierta **personalización** de los contenidos (Rost, 2003).

Periódico Granma

Granma es un periódico cubano, considerado como el Órgano Oficial del Comité Central del Partido Comunista de Cuba y fundado en el año 1965 por Juan Hernández Igarza. Su nombre proviene del yate que transportó a Fidel y a otros 81 rebeldes a las costas cubanas en el 1956, dando inicio a la Revolución Cubana. El periódico es publicado y actualizado todos los días y ampliamente leído dentro de la isla de Cuba. Existen varias ediciones internacionales, disponibles en español, inglés, francés, italiano, alemán y

portugués, todas ellas accesibles por Internet. El periódico está compuesto regularmente por discursos de líderes del gobierno cubano (incluyendo al Comandante en Jefe Fidel Castro), anuncios oficiales del gobierno cubano, relatos acerca de la historia de la lucha revolucionaria en Cuba desde el siglo XIX hasta la actualidad, noticias y comentarios de política latinoamericana y mundial, programación de la Televisión Cubana de cada día, suplementos especiales, entre otros (Periódico Granma, 2011).

Sistema de Gestión Editorial

El Sistema de Gestión Editorial para el Periódico Granma (SGEPG) es una aplicación informática sobre tecnología web que permite la correcta informatización del proceso de edición de materiales periodísticos en la sede del Periódico Granma. El mismo está basado en el Sistema de Gestión de Contenidos (CMS) Drupal liberado bajo licencia libre y desarrollado sobre el lenguaje PHP. En la solución fueron utilizados módulos desarrollados por la comunidad de desarrolladores del CMS. Posibilita la edición enriquecida de texto, permitiendo copiar textos desde otras fuentes de información. Este sistema tiene la posibilidad de incorporar automáticamente al flujo de producción los materiales que han arribado a la sede del periódico mediante correo electrónico mediante el uso de un formato o de una plantilla específica para este fin. Cuando un contenido ha sido revisado, corregido y aprobado se encontrará accesible para emplanar la edición impresa del periódico y a su vez podrá ser transmitido hacia la versión digital del periódico en Internet mediante el uso de estándares de transmisión de noticias. Los contenidos procesados por este sistema serán correctamente archivados para brindar un servicio de recuperación a sus usuarios (Aparicio, 2011).

Sistema de Gestión Documental

El Sistema de Gestión Documental de Fotografías para la Prensa es un conjunto de soluciones informáticas para la web, basada en estándares abiertos para la gestión de recursos fotográficos de los medios de prensa cubanos. Para ellos se propone utilizar una personalización de un ECM desarrollado sobre el lenguaje Java, como núcleo del sistema y un framework desarrollado sobre el lenguaje PHP, como interfaz web del sistema, ambos liberados bajo licencias libres. El SGDFP está orientado a la gestión de los recursos fotográficos, dígame fotografías, unidades documentales, galerías de imágenes y carpetas. El flujo de producción de dichos recursos comprende la selección, identificación, descripción, representación, recuperación, publicación y generación automática de diferentes perfiles a partir del

formato original que se ha incorporado al sistema. El sistema deberá integrarse con un grupo de herramientas empleadas en la redacción de las informaciones y el montaje de las noticias en los diferentes medios de publicación, como son: Sistema de Gestión Editorial y Plataforma de Publicación Web (Lissabet, 2011).

1.2 Tecnologías, herramientas y lenguajes de programación a investigar

1.2.1 Sistemas de Gestión de Contenidos (CMS)

Un Sistema de Gestión de Contenidos o CMS (*Content Management System*) es un programa para la gestión y publicación de un sitio web dinámico, con actualizaciones periódicas, apariencia uniforme y diseño centrado en el usuario (Juglar, 2007).

El Sistema de Gestión del Contenido es una herramienta para gestionar el contenido de una página web. Dentro de una página web se pueden distinguir dos partes:

Presentación: que incluye todo lo que forma el estilo, estructura y disposición de la página, como son: los ficheros HTML, hojas de estilos e imágenes.

Contenido: contiene los datos que los autores publican en la página, como son: texto y fotografías (García, 2004).

Los CMS son sistemas configurables que permiten la gestión de contenidos y la personalización de los mismos. El principal objetivo de estos es facilitar el desarrollo de aplicaciones web que manipulen los contenidos de forma dinámica por parte de usuarios que no tengan mucha experiencia en el trabajo con los mismos, agilizando el trabajo de desarrollo puesto que muchas funcionalidades ya están desarrolladas. Las funcionalidades más importantes que brindan son: la creación, gestión y publicación de contenidos.

Drupal

“Drupal es un Sistema de Gestión de Contenidos (CMS) modular, gratuito y de código abierto para todos” (Mercer, 2008). Dicho CMS permite crear páginas web de cualquier tipo, adaptándose a las necesidades particulares de cada caso, se demuestra esto en la gran cantidad de sitios web desarrollados bajo esta aplicación, que hoy se publican en Internet.

Drupal permite tener un control sobre el flujo de creación, edición, borrado y publicación de contenidos en diversos formatos, además de categorizar y organizar los mismos. “Es un sistema ampliable y extensible, presenta un sistema de plantillas (themes) separado de la lógica, y por defecto la biblioteca de JavaScript jQUERY” (Tomlinson y otros, 2007).

Brinda además un conjunto de módulos que proporcionan diferentes funcionalidades, algunos módulos ya los tiene en el núcleo del sistema, pero otros se le pueden añadir según la funcionalidad que desee agregar, los mismos se pueden descargar del repositorio oficial. Drupal posee una capa de abstracción de base de datos que actualmente está implementada para MySQL y PostgreSQL aunque permite adicionar con facilidad soporte a otras bases de datos (Gersbach, 2010).

Entre el núcleo y los módulos contribuidos Drupal le permite hacer disímiles cosas y si no tiene éxito en una función en especial, puede que el mismo se integre con un servicio que pueda prestar la función para desarrollar lo que necesita. “Un buen ejemplo de apoyo de servicios a terceros implica la integración con Google Analytics para las estadísticas de avance que tiene un sitio web. Drupal también se integra con los servicios anti-spam como Akismet y Mollom” (Safuto, 2011).

Este CMS está publicado bajo la licencia GNU/GPL 2, por lo que se puede descargar libremente, modificar y reutilizar el código del mismo. El mismo se enriquece con una gran comunidad de usuarios que se encarga de desarrollarlo y mantenerlo. Es multiplataforma, puede funcionar tanto en Apache, Microsoft IS, servidores web y en los sistemas operativos GNU/Linux, Windows, Mac OS X, BSD y Solaris. Es totalmente portable ya que esta implementado en el lenguaje de programación PHP (Gersbach, 2010).

La seguridad de un portal desarrollado con Drupal no requiere de módulo adicional, ya tiene en el núcleo un nivel de acceso para la gestión de usuarios que se basa en roles y permisos, garantizando la seguridad del sistema y la privacidad de los usuarios, aunque existen módulos que mejoran la seguridad del sistema. El CMS tiene diferentes niveles de caché que funcionan como mejoras y existen módulos que enriquecen este proceso, lo cual mejora el rendimiento de la aplicación web; uno de los sistemas con los que se integrará la plataforma está desarrollado con dicho CMS y el administrador no tendrá que capacitarse en varios CMS para trabajar con los sistemas.

En el caso del posicionamiento web, Drupal tiene ya implementados algunos módulos que por defecto hacen este trabajo como el “Path” que permite redefinir las url y hacerlas mucho más claras, otros módulos fueron creados por la comunidad y pueden ser descargados y actualizados en el sitio oficial del CMS.

WordPress

Es un sistema de gestión de contenidos enfocado a la creación de *blogs*. Entre las características principales de WordPress se encuentra que es de fácil instalación, actualización, personalización y uso en general, posee herramientas de comunicación entre *blogs*, sistemas de enlaces, calendarios, además de

contar con una variedad de *plugins* y temas adicionales. También permite que sea administrado por diferentes autores, comentarios que pueden ser publicados, borrados o marcados como spam, widgets para los temas, admite *plugins*, gestión y distribución de enlaces, y además permite ordenar artículos y páginas en categorías, subcategorías y etiquetas (GoldScripts, 2007)

Además de ser útil para la administración de *blogs*, es bueno para la creación de distintos tipos de sitios web, como una tienda en línea o una página corporativa de una empresa. Es una herramienta ideal para aquellos con pocos conocimientos de HTML, PHP y para los programadores avanzados que desean aprovechar el potencial de dicha herramienta.

1.2.2 Frameworks (CMF)

Los Frameworks apoyan en el desarrollo del *software*, proporcionando una mayor rapidez a partir de la estructura definida que presentan, brindando una gran organización en el desarrollo de la aplicación, garantizando así un correcto mantenimiento del sitio. “Los Frameworks son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos. Se utiliza la Programación Orientada a Objetos (POO), permitiendo la reutilización del código” (Pérez, 2007).

CodeIgniter

Es un framework sencillo para desarrollar sitios web basados en PHP. Sus requerimientos son pocos como PHP 4.3.2 y es compatible con varias bases de datos como MySQL (4.1+), MySQLi, MS SQL, Postgres, Oracle, SQLite, y ODBC. Actualmente está en su versión 2.1.0. Provee una serie de bibliotecas para funcionalidades comúnmente utilizadas así como una sencilla interfaz y estructura lógica de acceso a estas. Es ligeramente sencillo a diferencia de otros frameworks que necesitan muchas bibliotecas para funcionar, CodeIgniter se las arregla con pocas bibliotecas y si necesita de otros los añade dinámicamente. Utiliza el patrón modelo-vista-controlador (MVC) para fijar los contenidos por la eficacia que provee este patrón para separar la parte lógica de la presentación. CodeIgniter está bajo licencia libre Apache/BSD-style por lo que puede ser utilizada sin riesgo alguno (Upton, 2007).

CodeIgniter provee los códigos y bibliotecas más comunes para el desarrollo de aplicaciones web, sin embargo no cuenta con una comunidad de desarrollo tan grande que le aporte otras funcionalidades como las que poseen otros frameworks de desarrollo.

Symfony

Es un framework de licencia de *software* libre para el desarrollo de aplicaciones web. Trabaja basado en el patrón modelo-vista-controlador (MVC). Proporciona herramientas y clases para reducir el tiempo de desarrollo de aplicaciones web complejas y automatiza las tareas más comunes y ayudando al desarrollador a enfocarse en tareas más específicas. Está desarrollado en PHP5 y es compatible con la mayoría de los gestores de bases de datos como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Es multiplataforma, o sea, puede ser ejecutado tanto en plataformas *nix (unix, GNU/Linux, entre otros.) como en Windows. Actualmente se encuentra en su versión 2.0.x (Méndez, 2010).

Symfony es uno de los mejores frameworks existentes hasta el momento, con una gran comunidad que le brinda soporte continuo y añade funcionalidades que pueden ser descargadas gratuitamente desde su sitio oficial, es una de sus grandes ventajas por tener licencia de *software* libre; sin embargo, pese a estar dirigido al desarrollo de aplicaciones web en poco tiempo, es necesario tener sobre este un conocimiento avanzado y amplio dominio de sus funcionalidades como son su modo de tratar la gestión de datos. Symfony puede ser utilizado para todo tipo de aplicaciones web que necesiten desarrollar una infraestructura nueva desde su base y sean a largo plazo y por versiones. En cuestiones de desarrollar en poco tiempo y con pocos desarrolladores no es muy factible utilizarlo si no se tiene experiencia con este framework, consume demasiado tiempo de aprendizaje y para este tipo de proyecto no es necesario crear una infraestructura compleja y nueva, que Drupal ofrece por defecto como gestor de contenidos.

1.2.3 Gestores de base de datos

“Los gestores de base de datos no son más que programas no visibles al usuario final los cuales tienen como primicia fundamental encargarse de la privacidad, seguridad e integridad de los datos y su interacción con el sistema operativo” (Álvarez, 2007).

PostgreSQL

Es un gestor de bases de datos que implementa el mapeo objeto-relacional (ORM, se refiere al conjunto de objetos relacionados entre sí en las bases de datos y el mapeo, a la descripción y estructura de un objeto), de código abierto y de los sistemas gestores de bases de datos más potentes en el mercado. Utiliza el modelo multiprocesos para garantizar la estabilidad del sistema y en caso que exista un fallo en los procesos el sistema seguirá funcionando y los demás procesos no se afectarán.

PostgreSQL soporta casi todas las consultas SQL incluyendo subconsultas, transacciones y funciones definidas por el usuario y entre los lenguajes que tienen enlaces con este, se encuentran C, C++, Java, Perl, TCL y Python (Lockhart, 1999).

“Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (ha llegado a soportar el triple de carga de lo que soporta MySQL); implementa el uso de rollback's (es una operación que devuelve a la base de datos a algún estado previo), haciendo su funcionamiento mucho más eficaz; tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle” (Aliaga y Miani, 2011).

MySQL

Es un sistema de gestión de bases de datos relacional, multiusuario, de código abierto, se ofrece bajo la GNU/GPL para cualquier uso compatible con esta licencia. Proporciona un servidor de base de datos SQL (*Structured Query Language*) muy rápido (Axmark y Widenius, 2011). La principal función de este sistema es la velocidad y la robustez, para las columnas soporta gran cantidad de tipos de datos. Funciona sobre múltiples plataformas y sistemas operativos. Presenta un excelente nivel de seguridad en los datos y es fácil de configurar e instalar. Por su implementación multihilo, tiene un sistema flexible de gestión de usuarios y contraseñas.

“Carece de soporte para transacciones, *rollbacks* y subconsultas; no maneja la integridad referencial, haciendo de este gestor una solución pobre para muchos campos de la aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí poseen esta característica y no es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad” (Aliaga y Miani, 2011).

1.2.4 Servidor Web

Apache

Apache es un servidor web robusto y de los más usados en el mundo, presenta características como: gratuito, modular, de código abierto, rápido, eficiente y multiplataforma.

Apache Server 2.0 hace de Apache una solución web más flexible, más transportable y más escalable que nunca, esta versión introduce módulos multiprocesos.

Presenta mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Tiene un flexible sistema modular, permitiendo cargar y descargar módulos sin tener que tocar el kernel. Trabaja sobre múltiples plataformas, entre las que se encuentran: Unix, GNU/Linux, Mac OS X, Vms, Win32, OS2, entre otros; aunque trabaja más rápido y es más estable en BeOS, Windows y OS2. Incluye módulos que se cargan dinámicamente, soporta Perl, PHP, CGI; presenta soporte para bases de datos y soporte SS para transacciones seguras (ABCdatos, 2011).

“La API para los módulos ha cambiado significativamente en la nueva versión, muchos de los problemas de ordenación y prioridad de módulos de la versión 1.3 deben haber desaparecido. También se han añadido nuevas llamadas que ofrecen capacidades adicionales sin tener que parchear el núcleo del servidor Apache. Los mensajes de error que se envían a los navegadores están ahora disponibles en diferentes idiomas. Estos mensajes pueden personalizarse por el administrador del sitio web” (The Apache Software Foundation, 2011).

1.2.5 Administración de base de datos

pgAdmin3

El *software* pgAdmin3 es el máximo exponente de cliente gráfico de PostgreSQL, de código abierto, tiene licencia “Artist License”. La aplicación se puede utilizar en GNU/Linux, FreeBSD, Solaris, Mac OSX y Windows para gestionar PostgreSQL 9.1. En pgAdmin3 se puede trabajar con casi todos los objetos de la base de datos, examinar sus propiedades y administrar fácilmente los servidores PostgreSQL, las bases de datos, funciones, tablas, columnas, triggers, entre otros. Una característica interesante de pgAdmin3 es que, cada vez que se realiza alguna modificación en un objeto, escribe la(s) sentencia(s) SQL correspondiente(s), lo que hace que, además de una herramienta muy útil, sea a la vez didáctica.

pgAdmin3 también incorpora funcionalidades para realizar consultas, examinar su ejecución y trabajar con los datos. Todas estas características hacen de pgAdmin3 la única herramienta gráfica que se necesitará para trabajar con PostgreSQL, tanto desde el punto de vista del usuario como del administrador. Evidentemente, las acciones que se pueden realizar en cada momento vienen condicionadas por los permisos del usuario con el que se conecte a la base de datos (Camps Paré, 2007).

1.2.6 Lenguajes principales para el desarrollo

PHP Hypertext Pre-processor (PHP)

PHP es un lenguaje de programación orientado a la creación de páginas web. “Es un lenguaje script interpretado en el lado del servidor para generar páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el lado del servidor” (Álvarez, 2001).

Para que funcione el sitio PHP necesita estar montado sobre un servidor Apache o en un servidor Internet Information Server (IIS) con las bibliotecas de PHP. De manera general este lenguaje es fácil de aprender, se caracteriza por ser muy rápido, además presenta muchas funciones. Tiene una amplia capacidad de conexión con la mayoría de los manejadores de bases de datos MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros. “PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo GNU/Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más” (Briones y otros, 2009). Es libre por lo que es una opción muy fácil para todos, es multiplataforma y tiene la capacidad de expandir su potencial utilizando módulos.

The Extensible HyperText Markup Language (XHTML)

XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. XHTML surge como el lenguaje que tiene un etiquetado más estricto que HTML, permitiendo una correcta interpretación de la información.

XHTML, al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos en el código, entre ellos se encuentran: que dentro del documento debe haber una estructuración coherente donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados, entre otros (W3C, 2008).

Hojas de estilo en cascada - Cascade Style Sheets (CSS)

CSS es un lenguaje de hojas de estilos que está dirigido a controlar la presentación de los documentos digitales definidos con HTML y XML. “El uso de CSS brinda muchas ventajas frente al HTML puesto que además de permitir dar estilo más depuradamente, permite definir estilos a determinados grupos de elementos” (Eloi de San Martín, 2005).

El funcionamiento del CSS es mediante reglas, que son compuestas por un selector y declaraciones y se aplican a documentos HTML, el selector es la etiqueta HTML a la cual se le asigna el estilo y puede estar antecedida por un punto o una signo de número según el atributo que conforme esa etiqueta y las declaraciones son las propiedades y valores que se le asignan que ya vienen definidas en la versión del CSS. “El selector funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La declaración es la parte de la regla que establece cuál será el efecto” (Anon, 2008). Así queda conformado el estilo de una aplicación web, donde el desarrollador puede brindarles a los usuarios finales una interfaz agradable, con colores y estructura adecuados.

JavaScript

“JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas” (Eguíluz Pérez). A una aplicación web se le pueden adicionar una gran variedad de efectos que pueden hacerlo más atractivo y esto solo se puede hacer con JavaScript. Es un lenguaje interpretado. Se caracteriza por trabajar con bibliotecas que tienen funciones implementadas y que resuelven disímiles problemas relacionados con la web.

JavaScript es el lenguaje del lado del cliente que más se usa en la actualidad, ya se demuestra en Internet la gran cantidad de páginas web publicadas que adicionan elementos que lo usan. “La mayoría de los exploradores web pueden trabajar con él (aunque en el caso de Microsoft estemos hablando de JScript). Y tanto Microsoft como Netscape siguen investigando otras formas de sacar más provecho a este lenguaje” (Bradenbaugh, 2000). La mayoría de los usuarios de Internet pueden trabajar en este lenguaje por la cantidad de factores positivos que tiene JavaScript.

jQuery

jQuery es una biblioteca de JavaScript que aporta una serie de funciones que ya están probadas para realizar tareas habituales, se ejecutan correctamente y no necesita volver a programarlas. Una de las principales funcionalidades que brinda jQuery es cuando el programador está desarrollando alguna

aplicación web y necesita que los script sean compatibles en cualquier tipo de navegador que esté usando el cliente, ya que implementa una serie de clases (de programación orientada a objetos) que permiten programar sin preocuparse del navegador que está usando el usuario final. Esta biblioteca ofrece una infraestructura que facilita el desarrollo de complejas aplicaciones web, provee una gran ayuda a la hora de trabajar con efectos dinámicos y con aplicaciones que usen AJAX. Tiene licencia para uso en cualquier tipo de plataforma. Basta con incluir en una aplicación web un pedazo de script JavaScript que contenga código jQuery, para comenzar a trabajar y probarlo. El archivo de la biblioteca ocupa solo 56 KB, no retrasa mucho la carga del sitio web, el servidor lo enviará al cliente la primera vez que visite una página del portal y en siguientes páginas el cliente ya tendrá el archivo de la biblioteca cargado en la caché, por lo que no necesitará transferirlo (Álvarez, 2010).

El uso de AJAX (Asynchronous JavaScript And XML)

El término AJAX se puede traducir como “JavaScript asíncrono + XML”. AJAX es una mezcla de varias tecnologías independientes que se unen de manera sorprendente. “Las tecnologías que forman AJAX son las siguientes:

1. XHTML y CSS, para crear una presentación basada en estándares.
2. Document Object Model (DOM), para la interacción y manipulación dinámica de la presentación.
3. XML, XSLT y JSON, para el intercambio y la manipulación de información.
4. XMLHttpRequest, para el intercambio asíncrono de información.
5. JavaScript, para unir todas las demás tecnologías” (Eguíluz Pérez, 2008).

AJAX mejora la comunicación entre el usuario y la aplicación, impidiendo las recargas constantes a la aplicación, esto se realiza en un segundo plano, porque es el intercambio de datos con el servidor. Las aplicaciones implementadas con AJAX emplean una capa intermedia, entre el usuario y el servidor, que excluye la recarga constante de la aplicación mejorando así la respuesta del navegador, ya que nunca el usuario se encuentra con una página vacía que esté esperando por la respuesta del servidor web.

“Cada vez que el usuario realiza una acción que significa mostrar datos, la capa JavaScript, repite la acción anterior de manera invisible al usuario y muestra los datos deseados. Un excelente ejemplo de aplicación AJAX, bastante compatible entre navegadores es Google Maps, en ella se puede observar cómo se cambia la posición del mapa sin recargar la página” (Máster P. Intelectual, 2005).

1.2.7 Entorno de Desarrollo Integrado (IDE)

NetBeans

NetBeans IDE es un proyecto de código abierto que posee una gran cantidad de usuarios. Se caracteriza por su portabilidad ya que es compatible con la mayoría de los sistemas operativos como Windows, Mac, GNU/Linux y Solaris. Esta plataforma de aplicación permite crear aplicaciones complejas o sencillas, que independientemente del tipo que sean permite que se desarrollen con rapidez, es una herramienta que permite que los desarrolladores escriban, compilen, depuren y ejecuten proyectos. Soporta varios lenguajes, tales como: Java, C, C++, PHP, Ada, Groovy, PL/SQL, Python, Ruby y todos los relacionados con la web (HTML, CSS, JavaScript y XML).

“NetBeans IDE 7.0 introduce soporte para los lenguajes de programación propuestos en las especificaciones de Java SE 7 con el desarrollo previo de JDK 7. Esta versión también ha mejorado la integración con el servidor de Oracle WebLogic, así como el soporte para la base de datos Oracle y GlassFish 3.1. Además incluye Maven 3 y soporte para la edición de HTML5, un diseñador de GridBagLayout nuevo para mejorar el desarrollo GUI Swing, también trae mejoras en el editor de Java” (Parra, 2011). NetBeans IDE 7.0 está disponible en inglés, portugués brasileño, japonés, ruso y chino simplificado. Presenta una característica principal y es que mejora notablemente el rendimiento.

1.2.8 Servicio Web

Los servicios web según el W3C (*World Wide Web Consortium*) lo define como: “... un sistema de software diseñado para soportar interacción interoperable máquina a máquina sobre una red. Este tiene una interfaz descrita en un formato procesable por una máquina (específicamente WSDL). Otros sistemas interactúan con los servicios web en una manera prescrita por su descripción usando mensajes SOAP, típicamente enviados usando HTTP con una serialización XML en relación con otros estándares relacionados con la web” (W3C, 2004).

En otras palabras, no es más que la forma en que dos sistemas se comunican utilizando el Internet como medio de transporte de datos y XML como contenedor de la información. Siempre existiendo un lado servidor, que es donde se genera el servicio y un lado cliente que lo consume.

XML-RPC

Es un protocolo de llamada a procedimiento remoto que usa XML como codificador de los datos HTTP

como protocolo de transmisión de mensajes. Es un protocolo sencillo que define unos pocos tipos de datos y comandos útiles. Fue creado en 1998 por Dave Miller de la empresa UserLand Software en asociación con Microsoft; siendo tan sencillo, Microsoft decidió añadirle otras funcionalidades y se convirtió en algo más complejo y que hoy llamamos SOAP.

SOAP

(*Simple Object Access Protocol*) Es un protocolo de acceso que permite la comunicación entre aplicaciones remotas a través de Internet. Creado por Microsoft y actualmente bajo el auspicio de W3C. Ha sido el estándar básico utilizado para la construcción de servicios web a nivel mundial por mucho tiempo y es una evolución y mejora del antes mencionado XML-RPC. La diferencia distintiva de ambos protocolos es que en los procedimientos SOAP los parámetros tienen nombre y no interesa su orden, no así en la XML-RPC que sí importa.

REST

(*Representational State Transfer*) Según Fielding, es un estilo de arquitectura de *software* para sistemas hipermedias distribuidos tales como la World Wide Web.

En otras palabras es un conjunto de principios para el diseño de redes, que es utilizado comúnmente para definir una interfaz de transmisión sobre HTTP de manera análoga a como lo hace SOAP. Aunque REST como tal no es un estándar, posee un conjunto de estándares tales como HTML, URL, XML, GIF, JPG y tipos MIME.

1.2.9 Herramientas de desarrollo

NuSOAP

Es una biblioteca de herramientas para desarrollar servicios web en el lenguaje PHP. Se compone de un grupo de clases y funciones que facilitan el trabajo a los desarrolladores para entender de una manera más sencilla los servicios web. Permite el soporte tanto para la creación de Clientes (quienes consumen los servicios web) como para los Servidores (quienes los proveen).

Extensión SOAP de PHP5

Es una extensión desarrollada en el núcleo de PHP desde su versión 5 y posterior. Permite la creación

tanto de clientes y servidores de forma nativa en este lenguaje.

Zend SOAP Server

Herramienta para el desarrollo de servicios web mediante SOAP del Zend Framework. Este es de código abierto y provee clases y funciones que apoyan el desarrollo de servicios web de un modo sencillo. Está orientada a objetos y sus bibliotecas de clases y funciones permiten ser usadas independientemente ya que existe una baja dependencia entre sus componentes. Siendo su arquitectura tan débilmente acoplada se pueden utilizar sus componentes por separado, este tipo de diseño es conocido como "use at will" (Uso a voluntad). Los servicios web implementados con las bibliotecas de Zend Framework pueden funcionar con o sin WSDL, el cual permite generarlo de forma automática en el caso necesario.

Módulo Services 3.0

Módulo desarrollado por la comunidad de Drupal orientado a la gestión de servicios web tanto para brindarlos como para consumirlos. Permite crear servicios web mediante una interfaz y códigos propios de CMS. Funciona bajo múltiples interfaces tales como REST, XML-RPC, JSON, SOAP y otros (Drupal Association, 2010).

Módulo SOAP Server 3.0

Extensión del módulo Services desarrollada por la comunidad de Drupal, para proveer servicios mediante el uso de SOAP (Drupal Association, 2010).

1.2.10 Metodología de desarrollo de software

“Las metodologías de desarrollo de software son un conjunto de técnicas, procedimientos, herramientas y un soporte documental que ayudan a los desarrolladores a realizar un nuevo software” (Piattini, 1996). Persiguen tres necesidades: el desarrollo de mejores aplicaciones que conduzcan a una mejor calidad, que exista un proceso de desarrollo controlado de la aplicación y un proceso normalizado en una organización, no dependiente del personal. En el proceso de desarrollo de *software* la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo para obtener los distintos productos parciales y finales. En la actualidad entre las metodologías más usadas se encuentran: Proceso Unificado de

Racional (RUP), adaptable para proyectos de largo plazo y Programación Extrema (XP), el cual se recomienda para proyectos de corto plazo.

SXP (Scrum - eXtreme Programming)

SCRUM es una metodología de desarrollo y una forma de auto-gestión de los equipos de programadores, dichos equipos deciden la forma de desarrollar las tareas asignadas y que tiempo le van a dedicar a las mismas. La metodología apoya el trabajo en equipo, donde los integrantes trabajan como un todo, en una misma dirección y con un objetivo claro. Es una metodología especialmente indicada para pequeños equipos de desarrollo y se orienta a una entrega rápida de resultados y una alta flexibilidad. Se complementa con la metodología de desarrollo de *software* ágil como la Programación Extrema (XP), la cual es una de las más exitosas en la actualidad, utilizada para proyectos de corto plazo y equipos pequeños; consiste en una programación rápida o extrema, pues uno de los requisitos que para llegar al éxito del proyecto es tener al usuario final como parte del equipo.

SXP es una metodología compuesta por XP y SCRUM e incluye procedimientos ágiles que permiten actualizar los procesos de *software* para el mejoramiento del desarrollo productivo, se desarrolla una gran preocupación e interés por el trabajo, procurando la responsabilidad de los integrantes del equipo, y apoyando al líder del proyecto en el control del mismo.

El proceso de *software* contiene dos vertientes: La gestión de proyectos y la ingeniería de *software*. La primera utiliza las técnicas más significativas de SCRUM, una de ellas es la planificación de los proyectos ya que SCRUM es una forma de gestionar proyectos de software, no es una metodología de análisis, ni de diseño, es una metodología de gestión del trabajo y la segunda utiliza las características más destacadas de XP, para llevar a cabo un proceso de desarrollo del proyecto que garantice que el mismo sea efectivo y eficiente. XP plantea cómo se debe desarrollar el *software* y SCRUM evalúa día a día lo que se va haciendo. SXP consta de 4 fases principales:

- Planificación-Definición: donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo: donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- Entrega y puesta en marcha.
- Mantenimiento: donde se realiza el soporte para el cliente.

De cada una de ellas se despliegan 7 flujos de trabajo: concepción inicial, captura de requisitos, diseño con metáforas, implementación, prueba, entrega de la documentación, soporte e investigación, el cual se utiliza por el equipo de desarrollo cuando sea necesario, es decir, es un flujo que se puede mover y utilizarlo en cualquier parte del ciclo de vida del proyecto. De estos flujos se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, planificación de las iteraciones y las actividades que se van a desarrollar para lograr el producto final, las pruebas, además de las tareas necesarias para realizar las investigaciones documentando todo el proceso (Marsi Peñalver, 2008).

RUP (Rational Unified Process).

El Proceso Unificado Racional (habitualmente resumido como RUP) es un proceso de desarrollo de *software* y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. El proceso de *software* que propone RUP presenta tres características esenciales:

1. Dirigido por casos de uso.
2. Centrado en la arquitectura.
3. Es iterativo e incremental.

La metodología RUP incluye las cuatro etapas más importantes: **Inicio, Elaboración, Construcción y Transición**, en las que se realizan diferentes iteraciones en dependencia del proyecto que se esté desarrollando y en las que se hace mayor hincapié en las distintas actividades.

En las etapas anteriormente mencionadas se incluyen una variedad de actividades, como son: modelación del negocio, análisis y diseño, implementación, pruebas y distribución, entre otras. RUP cuenta con roles, que a la hora de asignar las tareas permite que se realice de forma organizada, según el rol que representa el trabajador.

La metodología es más adaptable para proyectos de largo plazo, por lo que el proyecto que se vaya a desarrollar debe tener esta característica (Córdova Díaz, 2009).

1.2.11 Herramienta de modelado visual

Visual Paradigm

Visual Paradigm para el Lenguaje Unificado de Modelado (UML) es una potente herramienta CASE UML, multiplataforma, es fácil de instalar y actualizar, y puede ser utilizado durante todo el ciclo de vida del desarrollo de *software*: inicio, construcción, pruebas y despliegue. Está diseñada para una amplia gama de usuarios, incluidos los ingenieros de *software*, analistas de sistemas, entre otros, que están interesados en la construcción de grandes sistemas de *software* de la balanza de forma fiable a través del uso de la aproximación orientada a objetos.

“Puede integrarse en los principales IDEs, estar disponible en múltiples plataformas y brinda múltiples versiones para cada necesidad. Ofrece entre otras cosas: entorno de creación de diagramas para UML 2.1, diseño centrado en casos de uso y enfocado al negocio que generan un *software* de mayor calidad, uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación, capacidades de ingeniería directa (versión profesional) e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo” (Sierra, 2008).

Conclusiones parciales

Como resultado al estudio realizado de los CMS y los Frameworks, se seleccionan los CMS ya que permiten la gestión de contenidos y esta es una funcionalidad primordial con la que debe contar la Plataforma, esto simplifica en cierto modo el trabajo de los desarrolladores al no tener que preocuparse por la implementación de este flujo de trabajo y centrarse en el desarrollo de otras funcionalidades.

Se selecciona el CMS Drupal en su versión 7.14 por las facilidades que brinda, entre las que se destacan, que permite la publicación y gestión de contenidos en diversos formatos, además de categorizar y organizar los mismos, tiene incluido jQuery en el núcleo y se posibilita el uso de AJAX, es modular, gratuito y de código abierto para todos, garantiza la privacidad de los usuarios y la seguridad del sistema por la gestión de usuarios basada en niveles de acceso que presenta, permite integrarle otras funcionalidades a través de módulos; presenta total independencia con el tipo de base de datos, posee una capa de abstracción de base de datos que actualmente está implementada para MySQL y PostgreSQL aunque permite adicionar con facilidad soporte a otras bases de datos; puede integrarse con otros servicios que puedan prestar funcionalidades para desarrollar lo que necesita. Es multiplataforma y es totalmente portable ya que está implementado en el lenguaje de programación PHP. Para el

rendimiento, Drupal cuenta con un nivel de caché que funciona como mejora y existen módulos que enriquecen este proceso. En el caso del posicionamiento web, Drupal tiene ya implementados algunos módulos que por defecto hacen este trabajo.

No se selecciona WordPress porque este CMS está orientado al desarrollo de *blogs*, lo que dificulta la personalización de los campos que tiene cada tipo de contenido que se gestionan en el Periódico digital Granma, además solo es compatible con el sistema de gestión de bases de datos MySQL.

Como sistema gestor de bases de datos se elige PostgreSQL 9.1 por ser un sistema seguro, libre, potente, tiene un mejor soporte por parte de la comunidad, estable, posee una gran escalabilidad, capaz de soportar una mayor cantidad de peticiones simultáneas, soporta el triple de carga de lo que soporta MySQL; implementa el uso de rollback's, haciendo su funcionamiento mucho más eficaz, permite almacenar procedimientos en la propia base de datos.

En vista a que se selecciona Drupal 7.14 como CMS de desarrollo y el mismo está desarrollado en PHP se utiliza PHP 5.3 como lenguaje del lado del servidor. Para la estructura del documento del portal se propone el uso de XHTML 1.0, brindando un estilo agradable mediante el uso de CSS 2.0 y para ofrecerle al usuario final un uso más atractivo de la aplicación se selecciona AJAX ya que mezcla diferentes tecnologías.

Como metodología de desarrollo de *software* se propone SXP, ya que es una metodología de desarrollo ágil que se adapta a este tipo de proyecto por sus características de corto plazo y nivel de dificultad, con respecto a la documentación que necesita, siendo mínima y la necesaria. Se escoge sobre RUP por ofrecer una solución rápida, eficiente y menos compleja, el proceso de *software* de esta metodología contiene dos vertientes: la gestión de proyectos y la ingeniería de *software*, donde en la primera utiliza las técnicas más significativas de SCRUM y la segunda utiliza las características más destacadas de XP, para llevar a cabo un proceso de desarrollo del proyecto. SXP es una metodología ágil que se caracteriza por mantener al cliente como parte del equipo de desarrollo.

Como herramienta para el modelado de diagramas se propone el Visual Paradigm 5.0 porque es una herramienta para el diseño de los múltiples artefactos necesarios para representar la información en las metodologías de desarrollo y ofrece diversas facilidades cuando se realizan los diagramas UML y de modelado de datos. Esta versión actual posee un entorno de creación de diagramas para UML 8.0.

Se selecciona Apache 2.0, ya que es un servidor web robusto, de código abierto, rápido, eficiente y multiplataforma.

SOAP ha sido y es uno de los protocolos más utilizados para el diseño y construcción de servicios web en el mundo y que cuenta con una amplia comunidad de desarrolladores. Es fácil de consumir y dado el tiempo que lleva de existencia es un proyecto robusto que cuenta con buen soporte en los lenguajes de desarrollo de aplicaciones web como PHP. Permite definir tipos específicos y estandariza la comunicación entre ambos lados de los sistemas dejando los datos bien definidos. La comunidad de Drupal proporciona módulos para la gestión y desarrollo de servicios web, siendo estos de un gran avance al contribuir directamente con la seguridad del sistema proporcionando interfaces manejables y sencillas para administrar los servicios y proveer los accesos necesarios al sistema de forma segura, son estos el módulo Services y su complemento SOAP Server.

Introducción

El presente capítulo muestra la propuesta de solución para la problemática expuesta, haciendo uso de la metodología ágil SXP y con la ayuda de algunos recursos de apoyo tales como diagramas UML. Se realizará un correcto levantamiento de requisitos para que las principales funcionalidades del sistema queden planteadas y documentadas. Se explicará además la arquitectura y el diseño del sistema a desarrollar.

2.1 Propuesta de solución

Para dar solución al proceso existente se ha decidido desarrollar una Plataforma de Publicación Web utilizando el CMS Drupal, permitiendo una correcta integración con el Sistema de Gestión Editorial y Sistema de Gestión Documental para realizar la publicación dinámica de los contenidos. Por lo antes explicado se puede deducir que el uso de este sistema garantizará el cumplimiento de los diferentes requisitos planteados.

Principales funcionalidades de la aplicación:

- Gestión de los contenidos a publicar en la plataforma.
- Gestión de las páginas estáticas encargadas de mostrar información adicional.
- Gestión de las galerías de imágenes y opiniones gráficas.
- Gestión de boletines para enviar desde el periódico los últimos contenidos publicados.
- Búsqueda básica y avanzada de contenidos.
- Acceso a documentos digitales con un resumen de la información de la edición impresa.
- Interacción con la plataforma por medio de encuestas.
- Mostrar información climatológica de las distintas regiones del país.
- Mostrar la cartelera televisiva de los distintos canales de la Televisión cubana.
- Integración con el Sistema de Gestión Editorial.

Funcionalidades más generales:

- Gestión de usuarios y roles dentro de la aplicación.

- Funciones de búsqueda.
- Sistema de impresión de información.
- Seguimiento de los contenidos más visitados.

2.2 Proceso de desarrollo del proyecto basado en SXP

Para una mayor eficiencia a la hora de desarrollar las tareas asignadas y llevar a cabo un proceso de desarrollo organizado y efectivo, se tomó como base la metodología ágil SXP, creando los siguientes artefactos: tabla de roles, historias de usuarios y algunos modelos auxiliares.

La siguiente tabla recoge los diferentes roles que intervendrán en el desarrollo del *software* y sus responsabilidades.

Rol	Descripción	Nombre
Analista	Responsable de desarrollar las historias de usuario y las pruebas funcionales para validar su implementación.	Rosana Mir Ricardo Valdivia
Diseñador	Responsable del diseño del sistema, de los prototipos de interfaces, las metáforas y supervisor del proceso de construcción.	Rosana Mir Ricardo Valdivia
Arquitecto	Se relaciona directamente con el analista y el diseñador ya que su trabajo se vincula con la estructura y el diseño del sistema.	Rosana Mir Ricardo Valdivia
Programador	Responsable de elaborar el código de las nuevas funcionalidades a implementar.	Rosana Mir Ricardo Valdivia
Encargado de pruebas	Ejecuta las pruebas y es responsable de las herramientas de trabajo para las mismas.	Eduardo Manuel Macías Sotolongo

Tabla 1. Planificación por roles.

2.3 Modelo de dominio

El modelo de dominio brinda los conceptos significativos para el dominio del problema y los objetos o componentes del mundo real, no específicamente del *software*, este puede ser modelado a través de un diagrama de clases de UML. Este modelo brinda una idea de forma general de los objetos que maneja una organización, es una manera correcta de comenzar a realizar el diseño de un sistema.

Para el desarrollo de la Plataforma de Publicación Web se decidió representar este modelo, ya que el proceso a llevar a cabo en el periódico Granma no es tan complejo como para desarrollar un modelo de negocio. A continuación se describen los diferentes objetos y relaciones que participan en el proceso.

Plataforma Web: Sistema que publicará todos los contenidos y que se compone de páginas.

Usuario: persona que gestionará los contenidos.

Administrador: usuario que gestionará los contenidos manualmente en la Plataforma Web.

Sistema de Gestión Editorial: usuario que gestionará los contenidos en la Plataforma Web a través de un servicio web.

Páginas: sección de la Plataforma Web que se compone de Galerías de imágenes, Opiniones gráficas y Contenidos.

Contenido: sección que contiene información relevante del periódico.

Categoría: elemento que categoriza a las Páginas y los Contenidos.

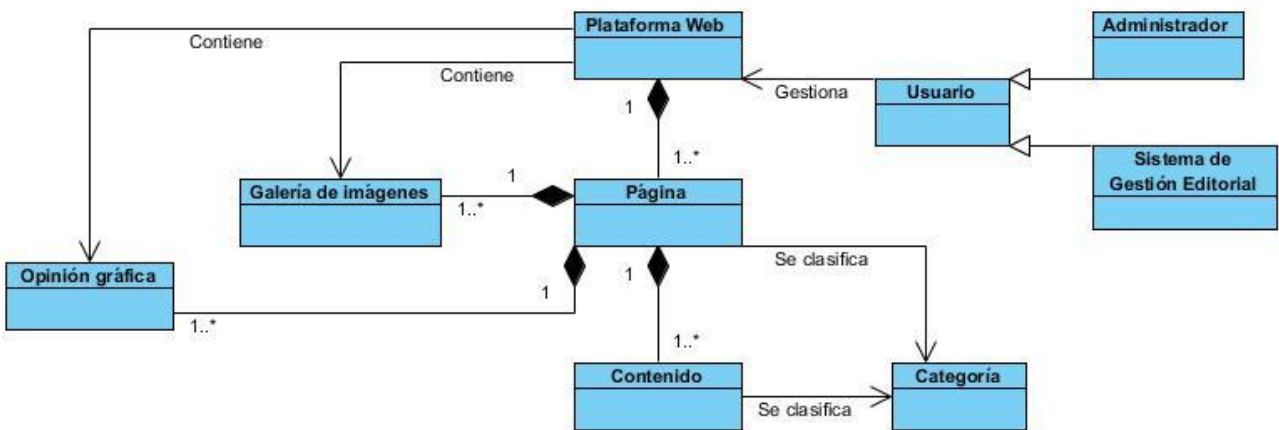


Figura 1. Modelo de dominio de la Plataforma de Publicación Web.

2.4 Levantamiento de Requisitos

El levantamiento de requisitos es de gran importancia en el desarrollo de un producto de *software*, a partir de estos se mide el éxito del proyecto. Para el levantamiento de requisitos es necesario conocer el entorno y las características de los usuarios que van a interactuar con el producto a desarrollar. Un requisito de un producto es una capacidad o característica que debe tener un sistema, que a partir de este se pretende solucionar una necesidad, y que es útil para los clientes. Existen dos tipos de requisitos,

funcionales y no funcionales, donde los funcionales son las condiciones que un producto debe cumplir y los no funcionales representan las cualidades que el producto debe tener.

2.4.1 Requisitos funcionales

RF1 – Insertar material periodístico.	RF36 – Mostrar edición impresa
RF2 – Modificar material periodístico.	RF37 – Insertar galería de imágenes
RF3 – Eliminar material periodístico.	RF38 – Modificar galería de imágenes
RF4 – Mostrar material periodístico	RF39 – Eliminar galería de imágenes
RF5 – Insertar páginas	RF40 – Mostrar galería de imágenes
RF6 – Modificar páginas	RF41 – Insertar opinión gráfica
RF7 – Eliminar páginas	RF42 – Modificar opinión gráfica
RF8 – Mostrar páginas	RF43 – Eliminar opinión gráfica
RF9 – Insertar correspondencia	RF44 – Mostrar opinión gráfica
RF10 – Modificar correspondencia	RF45 – Insertar encuesta
RF11 – Eliminar correspondencia	RF46 – Modificar encuesta
RF12 – Mostrar correspondencia	RF47 – Eliminar encuesta
RF13 – Insertar estado del tiempo	RF48 – Mostrar encuesta
RF14 – Modificar estado del tiempo	RF49 – Insertar enlaces
RF15 – Eliminar estado del tiempo	RF50 – Modificar enlaces
RF16 – Mostrar estado del tiempo	RF51 – Eliminar enlaces
RF17 – Insertar pancartas publicitarias	RF52 – Mostrar enlaces
RF18 – Modificar pancartas publicitarias	RF53 – Insertar síguenos en
RF19 – Eliminar pancartas publicitarias	RF54 – Modificar síguenos en
RF20 – Mostrar pancartas publicitarias	RF55 – Eliminar síguenos en
RF21 – Insertar cartelera	RF56 – Mostrar síguenos en
RF22 – Modificar cartelera	RF57 – Insertar especial
RF23 – Eliminar cartelera	RF58 – Modificar especial
RF24 – Mostrar cartelera	RF59 – Eliminar especial
RF25 – Insertar efemérides	RF60 – Mostrar especial
RF26 – Modificar efemérides	RF61 – Brindar servicio web para la inserción

RF27 – Eliminar efemérides	de contenidos
RF28 – Mostrar efemérides	RF62 – Insertar contenido mediante archivo
RF29 – Insertar boletín	RF63 – Mostrar listado de contenidos periodísticos por categoría
RF30 – Eliminar boletín	RF64 – Mostrar lo más visto
RF31 – Suscribir al boletín	RF65 – Mostrar calendario
RF32 – Dar de baja del boletín	RF66 – Mostrar artículos relacionados
RF33 – Insertar edición impresa	RF67 – Mostrar conozca a Cuba
RF34 – Modificar edición impresa	RF68 – Mostrar quiénes Somos
RF35 – Eliminar edición impresa	

Tabla 2. Listado de Requisitos Funcionales.

2.4.2 Requisitos no funcionales

2.4.2.1 Usabilidad

RNF1 – Los usuarios que accedan a la Plataforma pueden tener sexo femenino o masculino, de cualquier edad, nivel educacional, ocupación, experiencia profesional y puede no tener alguna experiencia con la aplicación.

RNF2 – El tiempo que pueden llevar los usuarios anónimos en adaptarse a la aplicación es de un día y para el usuario administrador dos días.

RNF3 - Los menús permitirán una navegación sencilla, tanto a los usuarios con conocimientos avanzados de informática como a los usuarios más inexpertos.

RNF4 - Se proporcionará a los usuarios el manual de usuario y toda la documentación necesaria para lograr un entendimiento total de las acciones a realizar en el sistema.

RNF5 – El sistema debe presentar las siguientes características:

Ambiente

- Servidor web: Apache en su versión 2.0.
- Lenguaje de programación: PHP en su versión 5.3.
- El intérprete de aplicaciones web: Navegador.
- Sistema gestor de bases de datos: Postgres.

Hardware mínimo

- Un servidor de gama media a alta.

- 8GB de memoria RAM.
- Discos duros rápidos en el orden de los 10KRPM.
- Al menos dos procesadores de doble núcleo, que tengan fuente redundante.
- Tarjeta de red a 1GB Ethernet.

Hardware ideal

- 2 servidores con las características anteriormente mencionadas, uno para alojar al Drupal con el apache y otro para alojar la base datos con PostgreSQL.

2.4.2.2 Interfaz

RNF6 – El sistema debe ser compatible con los navegadores Mozilla Firefox en la versión 3.6 o superior, Internet Explorer en la versión 6 o superior, Safari, Opera y Chrome.

2.4.2.3 Portabilidad

RNF7 – El sistema debe ser multiplataforma, principalmente compatible con los sistemas operativos más usados, como son: Windows y GNU/Linux.

2.4.2.4 Seguridad

RNF8 – Ante cualquier falla en el sistema, se deben mostrar los errores sin dar detalles de información, que pudiera comprometer la seguridad e integridad del mismo.

RNF9 – El sistema debe estar disponible el 98% del tiempo, donde 2% restante se debe dedicar al soporte, actualización y corrección de errores.

RNF10 - Debe contarse con un sistema de salvacopias externas de la información, para casos de desastres.

2.4.2.5 Eficiencia

RNF11 - Establecer como línea base la rapidez y eficiencia de las operaciones, tanto de acceso a datos como al mostrar la información.

RNF12 - La plataforma debe presentar un bajo nivel de carga para asegurar un tiempo de respuesta rápido a las peticiones de los lectores.

2.4.2.6 Soporte

RNF13 - El sistema debe dar la posibilidad de ser mejorado y de incorporarle nuevos servicios en caso de ser necesarios.

2.4.2.7 Requisitos de Licencia

- Uso de la licencia Apache Software License (para versiones anteriores a 2.0) para el servidor web Apache.
- Uso de la licencia GNU/GPL para el CMS Drupal.
- Uso de la licencia BSD (*Berkeley Software Distribution*) para el SGBD PostgreSQL.
- Uso de la licencia PHP License para el lenguaje de programación PHP.

Todas consideradas licencias libres y flexibles aprobadas por la Fundación de Software Libre o FSF (*Free Software Foundation*), organización que se dedica a eliminar las restricciones sobre la copia, redistribución, entendimiento y modificación de programas de computadoras, con el objetivo de promocionar el desarrollo y uso del *software* libre en todas las áreas de la computación, pero muy particularmente, ayudando a desarrollar el sistema operativo GNU.

La Lista de reserva del producto con las estimaciones de desarrollo para cada requisito se ubica en el [Anexo 1](#).

Para un desarrollo ágil y un mejor entendimiento de las historias de usuario, algunos requisitos funcionales se agruparon. Las historias de usuario de la 1 hasta la 7 y de la 10 hasta la 16 están compuestas cada una por 4 requisitos que representan las operaciones de Insertar, Modificar, Eliminar y Mostrar. En la siguiente tabla se especifica a que Historia de usuario (HU) pertenece cada requisito.

HU	1	2	3	4	5	6	7	8	9	10	11
Requisitos	1-4	5-8	9-12	13-16	17-20	21-24	25-28	29-30	31-32	33- 36	37-40
HU	12	13	14	15	16	17	18	19	20	21	22
Requisitos	41-44	45-48	49-52	53-56	57-60	61	62	63	64	65	66
HU	23	24									
Requisitos	67	68									

Tabla 3. Requisitos por Historia de usuario.

2.5 Historias de Usuario

Para el desarrollo del proyecto se decidió utilizar la metodología ágil SXP, por lo que se sustituye la descripción de los casos de uso (artefacto generado por la metodología RUP) por las historias de usuario (técnica utilizada en XP para especificar los requisitos del *software*), que describen las tareas que en el sistema se deben realizar y que dependen de las especificaciones de los clientes. Deben ser escritas en lenguaje natural, sin un formato predeterminado, no excediendo su tamaño de unas pocas líneas de texto. Las historias de usuario guían la construcción de las pruebas de funcionalidad y se utilizan para estimar tiempos de desarrollo. A continuación se describen algunas de las historias de usuario con prioridad Muy Alta. Las tareas ingenieriles pertenecientes a cada una de las siguientes historias de usuario se encuentran en el [Anexo 2](#).

Historia de Usuario	
Número: HU-1	Nombre Historia de Usuario: Gestionar material periodístico.
Modificación de Historia de Usuario Número: 1	
Usuario: Rosana Mir Chávez, Ricardo Valdivia Freyre	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 0.8
Riesgo en Desarrollo: 1, 2, 3, 5, 7, 8, 9 y 10.	Puntos Reales: 0.8
Descripción: El usuario Inserta, modifica, muestra o elimina los contenidos de tipo Material Periodístico en las interfaces proporcionadas por la plataforma para la administración.	
Observaciones: Responde a los Requerimientos Funcionales 1, 2, 3 y 4.	

Historia de Usuario	
Número: HU-6	Nombre Historia de Usuario: Gestionar cartelera.
Modificación de Historia de Usuario Número: 1	

Usuario: Rosana Mir Chávez, Ricardo Valdivia Freyre	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1
Riesgo en Desarrollo: 1, 2, 3, 4, 5, 7, 8, 9 y 10.	Puntos Reales: 1
Descripción: El usuario Inserta, modifica, muestra o elimina los contenidos de tipo Cartelera en las interfaces proporcionadas por la plataforma para la administración.	
Observaciones: Responde a los Requerimientos Funcionales 21, 22, 23 y 24.	

Historia de Usuario	
Número: HU-10	Nombre Historia de Usuario: Gestionar edición impresa
Modificación de Historia de Usuario Número: 1	
Usuario: Rosana Mir Chávez, Ricardo Valdivia Freyre	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1
Riesgo en Desarrollo: 1, 2, 3, 4, 5, 7, 8, 9 y 10.	Puntos Reales: 1
Descripción: El usuario Inserta, modifica, muestra o elimina los contenidos de tipo Edición Impresa en las interfaces proporcionadas por la plataforma para la administración.	
Observaciones: Responde a los Requerimientos Funcionales 33, 34, 35 y 36.	

En la sección **Riesgo en desarrollo** se especifican los riesgos que inciden en la historia de usuario, estos riesgos se describen en la Lista de riesgo en el [Anexo 3](#).

Las restantes historias de usuario y las tareas ingenieriles se describen en el [Anexo 2](#).

2.6 Visión general de la arquitectura

Para desarrollar una aplicación web utilizando el CMS Drupal se necesita un sistema operativo que sirva de base para gestionar todos los recursos. Entre los recursos necesarios se debe contar con un servidor web que conjuntamente con un lenguaje de programación y un sistema gestor de bases de datos hagan posible el funcionamiento de dicha aplicación.

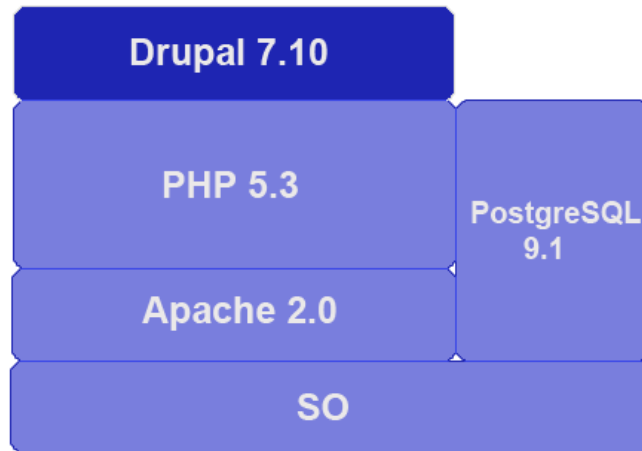


Figura 2. Arquitectura del Sistema donde está ubicado el CMS Drupal.

La arquitectura de *software* de Drupal está estructurada en capas, dicha arquitectura se asemeja al patrón modelo-vista-controlador.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el sistema de gestión de base de datos y la lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

En la figura 3, se muestra como está estructurado en capas el CMS Drupal, manteniendo la organización y flexibilidad del sistema, donde la vista son los Temas, Motor de Temas y JavaScript/AJAX; el modelo es la base de datos (BD), y el controlador es el Núcleo de Drupal y los Módulos; donde se hace uso de funciones (ganchos) para facilitar la comunicación entre el Núcleo de Drupal y los Módulos.

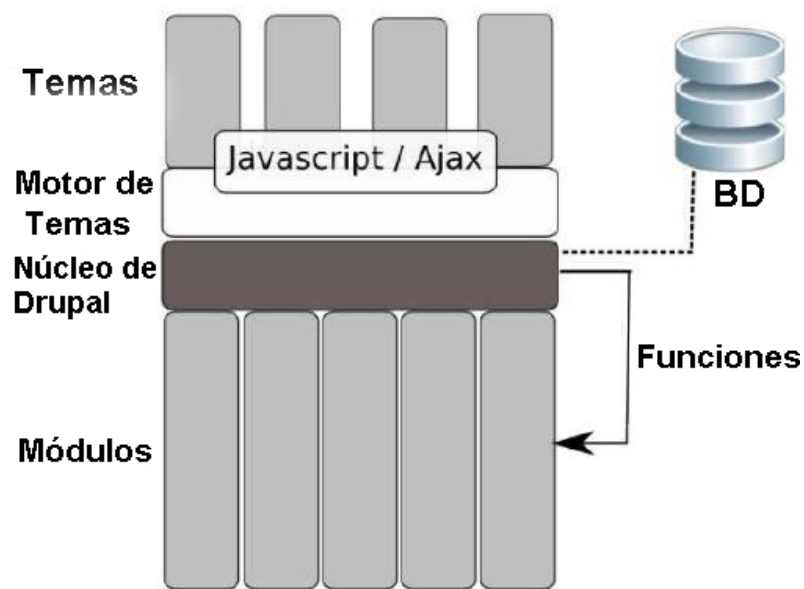


Figura 3. Arquitectura de Drupal.

Drupal utiliza algunas de las características de la Programación Orientada a Objetos, sin embargo, en su código no contiene la palabra *class*. El diseño de Drupal está altamente conformado a través de una orientación a objetos, mostrando características como: objetos y abstracción, además del encapsulamiento, polimorfismo y herencia, siendo estas tres últimas características las que hacen únicos a los Sistemas Orientados a Objetos. A continuación se demuestra la forma en que estas características están presentes en el CMS Drupal:

Objetos: algunos de los más prominentes componentes de Drupal que pueden ser considerados objetos, son los módulos, temas, nodos y usuarios.

Abstracción: el sistema de ganchos de Drupal (*hook system*) es la base de su interfaz de abstracción.

Encapsulamiento: dado que el código de Drupal está basado en funciones y todas comparten el mismo *namespace*, este está dividido por el uso de prefijos. Siguiendo esta simple convención, cada módulo puede declarar sus propias funciones y variables sin la preocupación de los conflictos con los demás.

Polimorfismo: los nodos son polimórficos en el sentido clásico. La representación real de un nodo, sin embargo, dependerá de qué tipo de nodo se pasa a la función, que es directamente análoga a tener un objeto de la clase y determinar su comportamiento cuando se envía un mensaje a ella.

Herencia: los módulos y los temas pueden definir la función que deseen, pero a la vez pueden heredar su comportamiento. En los temas, si no sobrescriben una función definida, se utiliza la que posee por defecto.

Entre los patrones de diseño que utiliza Drupal se encuentran: *observer*, *bridge* y *chain of responsibility*.

Observer (observador): En los módulos de Drupal se implementan los llamados *hooks* (ganchos), los cuales se comportan como observadores, estos esperan a que se ejecute un evento en el sistema para realizar el trabajo que le corresponde a cada uno.

Bridge (puente): Muchas de las funciones implementadas en los módulos de Drupal son desarrolladas sin pensar en el tipo de base de datos a usar en el sistema. Ejemplo de ellos son las funciones que están implementadas para crear la tablas donde se almacenaría la información de Drupal, para ello no se necesita saber qué tipo de base de datos se usará.

Chain of responsibility (cadena de responsabilidad): Para cada solicitud de una página en el sistema de Drupal, el sistema de menú determina si existe un módulo que atienda esta solicitud, si el usuario tiene acceso al recurso solicitado y que función fue llamada para realizar el trabajo.

Se hace uso además, del patrón de distribución Cliente-Servidor, el cual define una estructura para el modelo de despliegue, de uso muy extendido en aplicaciones web.

Los servicios web en la plataforma serán implementados con las funcionalidades nativas de PHP5 para SOAP; esta provee las funcionalidades estándares para el desarrollo de un servicio web sin necesidad de bibliotecas externas o frameworks, tanto para el cliente como el servidor. Las comunicaciones estarán aseguradas mediante autenticación de usuario, cada vez que se desee realizar una petición al servidor se comprobará que el usuario y la contraseña sean válidos en el sistema y tengan los permisos necesarios en la plataforma para utilizar el servicio web requerido, además se enviarán los mensajes de forma segura utilizando SSL.



Figura 4. Arquitectura de Servicios Web.

2.7 Diseño

Modelo de Diseño

Es un modelo de objetos, que describe la realización física de las historias de usuario, centrándose en cómo los requisitos funcionales y no funcionales, tienen impacto en la aplicación a desarrollar. Facilita la abstracción de la implementación del sistema y es de ese modo, el artefacto fundamental de entrada de las actividades de implementación.

2.7.1 Diagrama de paquetes

El diagrama que se muestra a continuación es el diagrama de paquetes que se compone de los paquetes CMS Drupal y Plataforma de Publicación Web, donde el paquete CMS Drupal contiene los paquetes que genera el CMS Drupal, que se tomará como base para la representación de las clases del diseño de la aplicación que se modela y el paquete Plataforma de Publicación Web contiene los paquetes que genera cada una de las funcionalidades a implementar en el sistema.

Para una mejor comprensión de los diagramas de clases de diseño, se decidió describir los paquetes que genera el CMS Drupal.

Include: Contiene un conjunto de ficheros imprescindibles para el funcionamiento, como el fichero Database, que provee las funcionalidades de acceso a la base de datos del CMS Drupal.

Theme: Contiene las plantillas de diseño, cuando se desee incluir uno nuevo, se copia la plantilla dentro de esta carpeta.

Misc: Contiene un conjunto de ficheros necesarios orientados al aspecto visual, CSS y JavaScript para el correcto funcionamiento de la aplicación.

Modules: Comprende todos los módulos, que permiten las distintas funcionalidades del CMS, cuando se desee incorporar un nuevo módulo, se tiene que copiar en dicha carpeta.

Luego de explicado el paquete CMS Drupal, se muestra el diagrama de paquetes donde el paquete Plataforma de Publicación Web está relacionado con el núcleo del mismo.

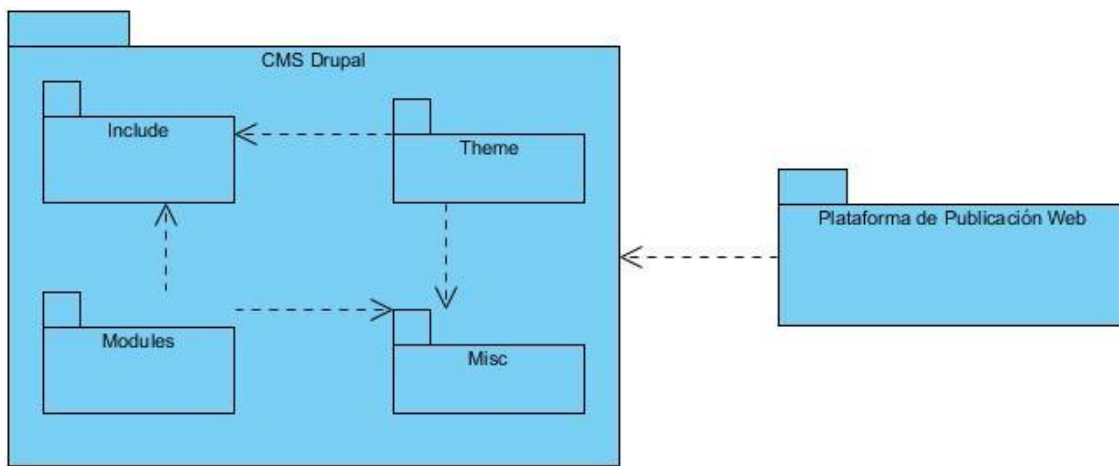


Figura 5. Diagrama de paquetes.

En la siguiente figura se muestran las funcionalidades que contiene la Plataforma de Publicación Web.

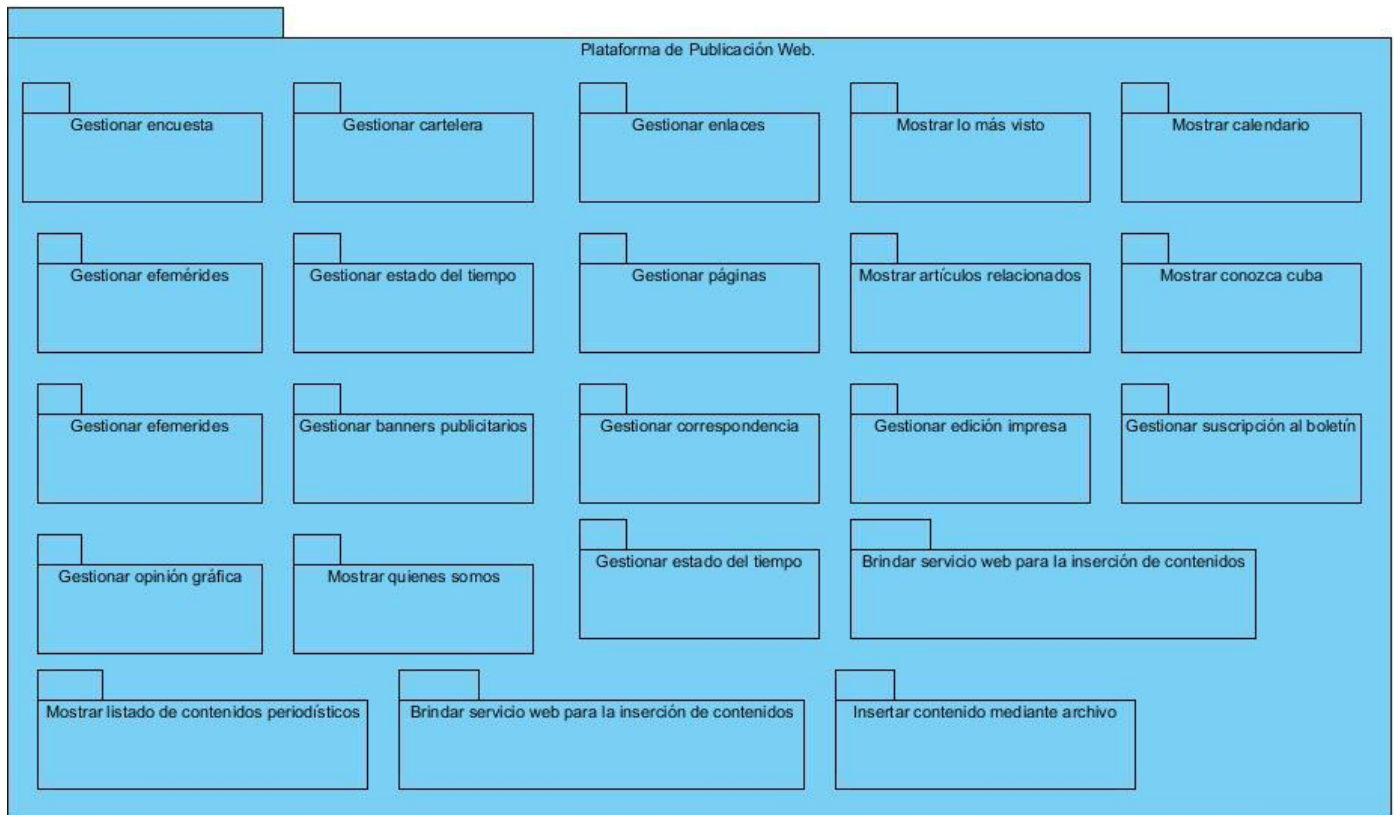


Figura 6. Diagrama de paquetes Plataforma de Publicación Web.

A continuación se muestran los diagramas de clases de diseño con estereotipos web de algunas historias de usuario con prioridad Muy Alta. Los restantes diagramas se ubican en el [Anexo 4](#).

Paquete Gestionar material periodístico

Para insertar un material periodístico la página servidora **SP_Gestionar material periodístico** construye la página cliente **CP_Insertar material periodístico**, la última se compone de un formulario **Form_Insertar material periodístico** donde se van a introducir los datos y que va a ser ejecutado por la página servidora. Para modificar un material periodístico la página servidora construye la página cliente **CP_Modificar material periodístico**, la última se compone de un formulario **Form_Modificar material periodístico** donde se van a modificar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora **CC_Gestionar material periodístico**. Para eliminar un material periodístico la página servidora construye la página cliente **CP_Eliminar material**

periodístico, la última se compone de un formulario **Form_Eliminar material periodístico** donde se van a eliminar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora. Para mostrar un material periodístico la página servidora construye la página cliente **CP_Mostrar material periodístico**, la última se compone de un formulario **Form_Mostrar material periodístico** donde se van a mostrar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora.

El paquete Clases persistentes de material periodístico se encuentra en el [Anexo 5](#).

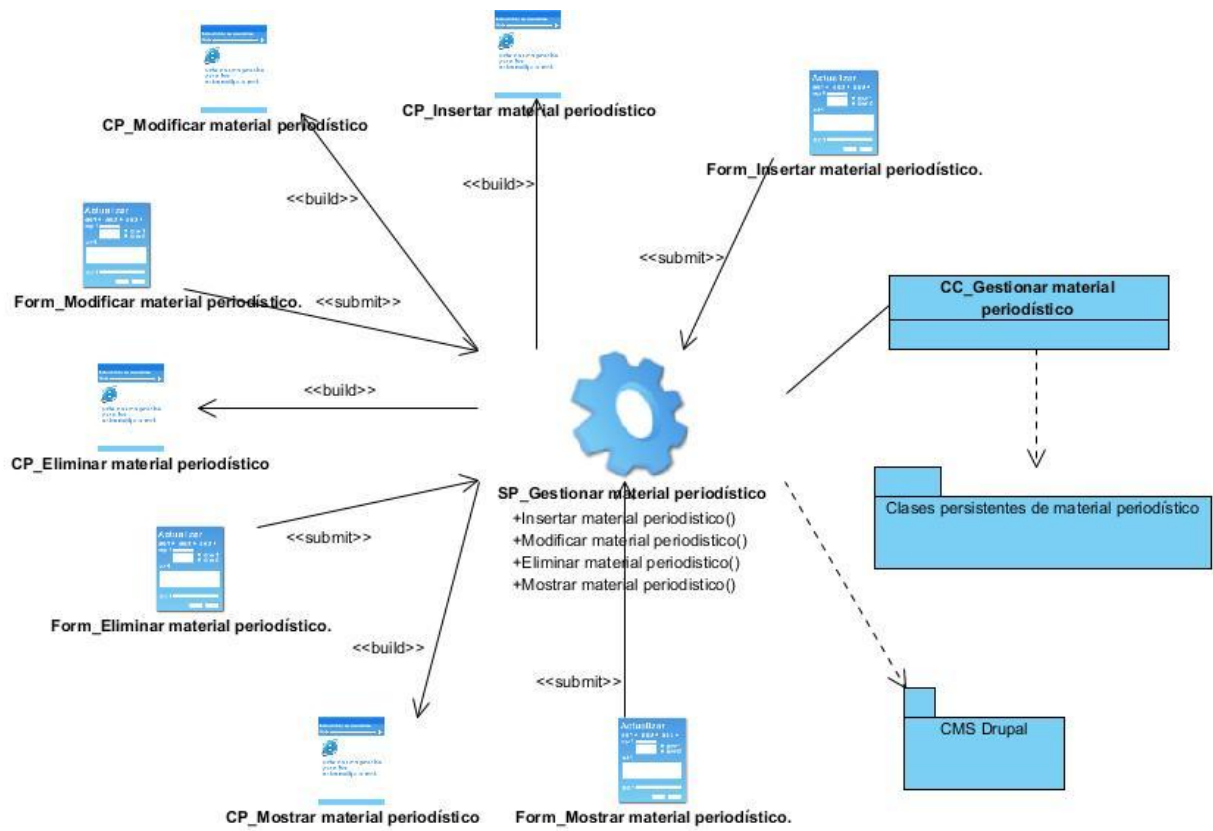


Figura 7. Paquete Gestionar material periodístico.

Paquete Gestionar cartelera

Para insertar una cartelera la página servidora **SP_Gestionar cartelera** construye la página cliente **CP_Insertar cartelera**, la última se compone de un formulario **Form_Insertar cartelera** donde se van a

introducir los datos y que va a ser ejecutado por la página servidora. Para modificar una cartelera la página servidora construye la página cliente **CP_Modificar cartelera**, la última se compone de un formulario **Form_Modificar cartelera** donde se van a modificar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora **CC_Gestionar cartelera**. Para eliminar una cartelera la página servidora construye la página cliente **CP_Eliminar cartelera**, la última se compone de un formulario **Form_Eliminar cartelera** donde se van a eliminar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora. Para mostrar una cartelera la página servidora construye la página cliente **CP_Mostrar cartelera**, la última se compone de un formulario **Form_Mostrar cartelera** donde se van a mostrar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora.

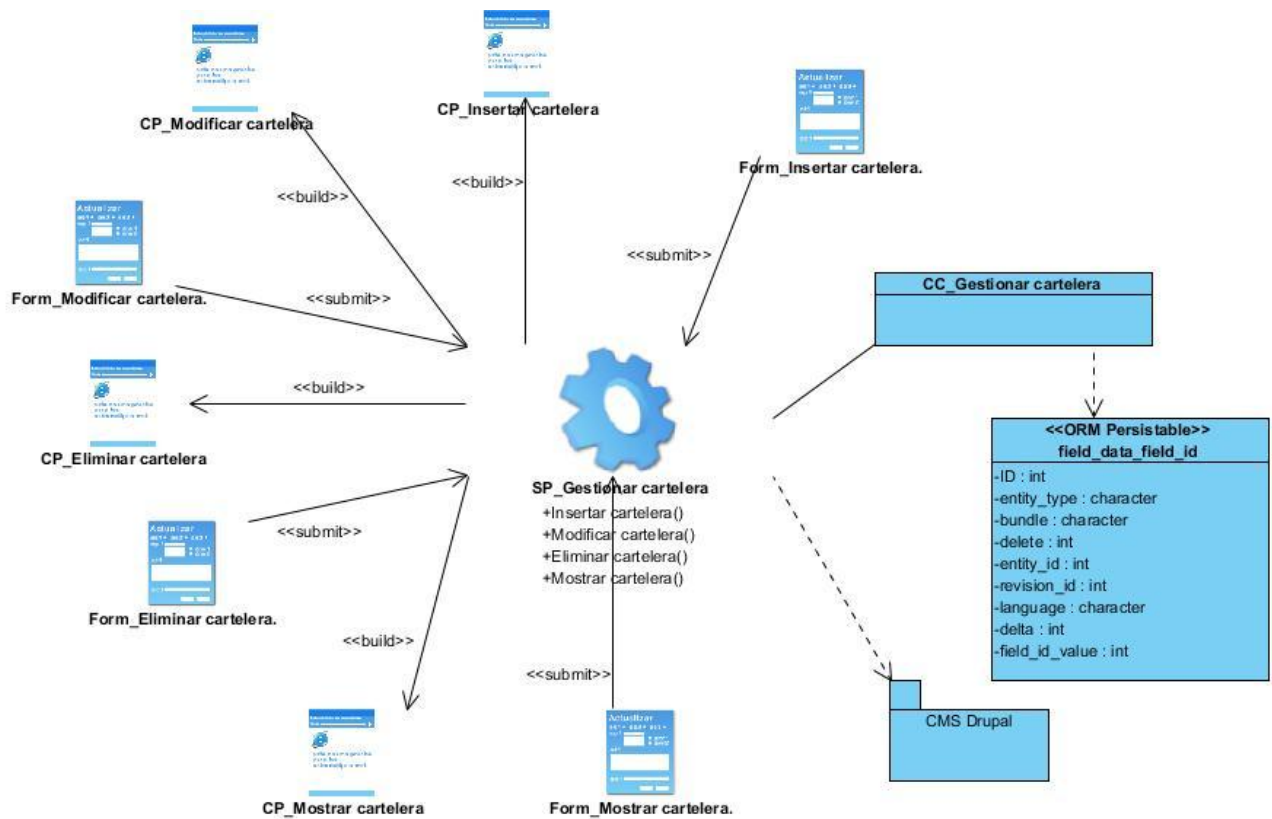


Figura 8. Paquete Gestionar cartelera.

Paquete Gestionar edición impresa

Para insertar una edición impresa la página servidora **SP_Gestionar edición impresa** construye la página cliente **CP_Insertar edición impresa**, la última se compone de un formulario **Form_Insertar edición impresa** donde se van a introducir los datos y que va a ser ejecutado por la página servidora. Para modificar una edición impresa la página servidora construye la página cliente **CP_Modificar edición impresa**, la última se compone de un formulario **Form_Modificar edición impresa** donde se van a modificar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora **CC_Gestionar edición impresa**. Para eliminar una edición impresa la página servidora construye la página cliente **CP_Eliminar edición impresa**, la última se compone de un formulario **Form_Eliminar edición impresa** donde se van a eliminar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora. Para mostrar una edición impresa la página servidora construye la página cliente **CP_Mostrar edición impresa**, la última se compone de un formulario **Form_Mostrar edición impresa** donde se van a mostrar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora.

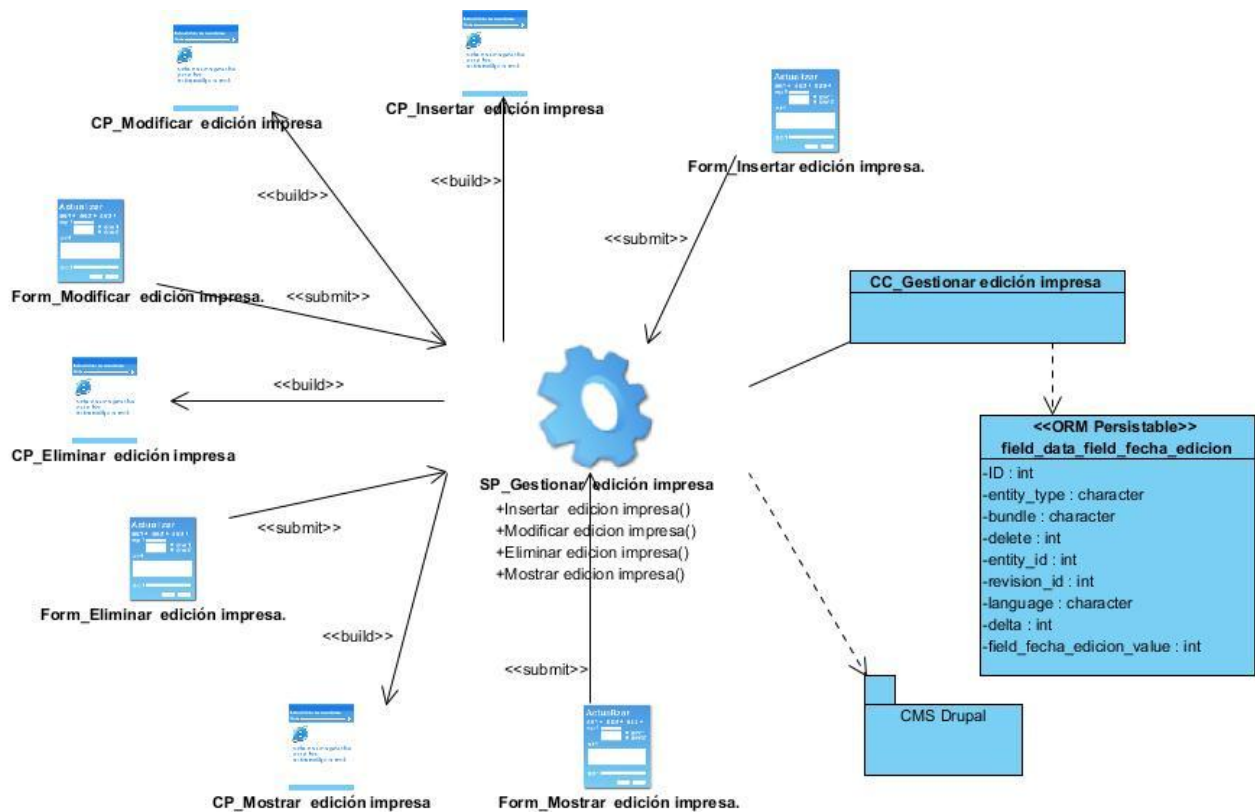


Figura 9. Paquete Gestionar edición impresa.

Paquete Gestionar galería de imágenes

Para insertar una galería de imágenes la página servidora **SP_Gestionar galería de imágenes** construye la página cliente **CP_Insertar galería de imágenes**, la última se compone de un formulario **Form_Insertar galería de imágenes** donde se van a introducir los datos y que va a ser ejecutado por la página servidora. Para modificar una galería de imágenes la página servidora construye la página cliente **CP_Modificar galería de imágenes**, la última se compone de un formulario **Form_Modificar galería de imágenes** donde se van a modificar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora **CC_Gestionar galería de imágenes**. Para eliminar una galería de imágenes la página servidora construye la página cliente **CP_Eliminar galería de imágenes**, la última se compone de un formulario **Form_Eliminar galería de imágenes** donde se van a eliminar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora. Para mostrar una galería de imágenes la página servidora construye la

página cliente **CP_Mostrar galería de imágenes**, la última se compone de un formulario **Form_Mostrar galería de imágenes** donde se van a mostrar los datos y que va a ser ejecutado por la página servidora; para acceder a los datos se hace mediante la clase controladora.

El paquete Clases persistentes de galería de imágenes se encuentra en el [Anexo 5](#).

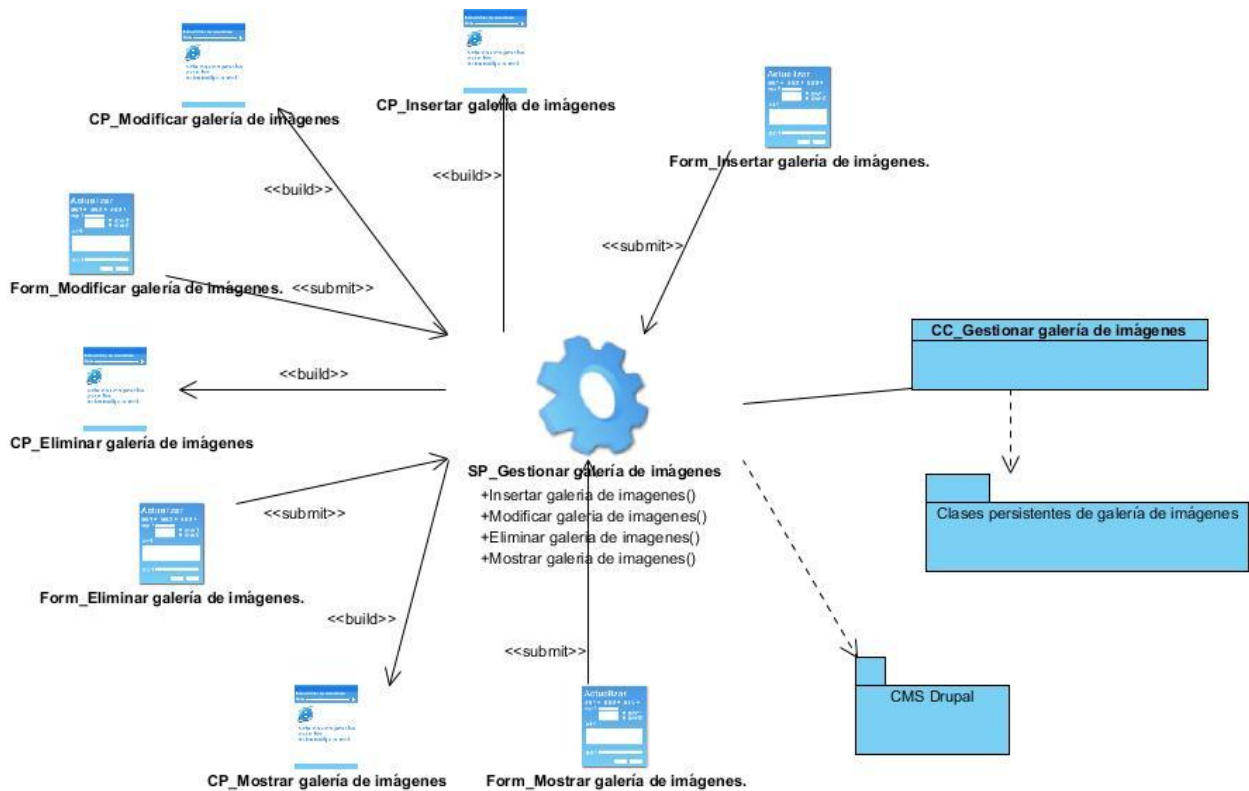


Figura 10. Paquete Gestionar galería de imágenes.

2.7.2 Diseño de la base de datos

“Las metas más importantes que se persiguen al diseñar un modelo de base de datos, es dividir la información en tablas organizadas en temas para reducir los datos redundantes, proporcionar el acceso a la información necesaria para reunir la información de las tablas cuando así se precise, ayudar a garantizar la exactitud e integridad de la información y satisfacer las necesidades de procesamiento de los datos y de generación de informes” (Vargas, 2005). Para diseñar la base de datos es necesario primeramente presentar el diagrama de clases persistentes con sus atributos y relaciones. Es necesario aclarar que el CMS Drupal cuenta con su propia base de datos, por lo que a continuación se presenta el

modelo de clases persistentes, a partir de las entidades más significativas que se le adicionaron a la Plataforma de Publicación Web, todas estas entidades se relacionan con la entidad node, ya que todos los contenidos que se crean en la plataforma se tratan como nodos.

2.7.2.1 Modelo de clases persistentes

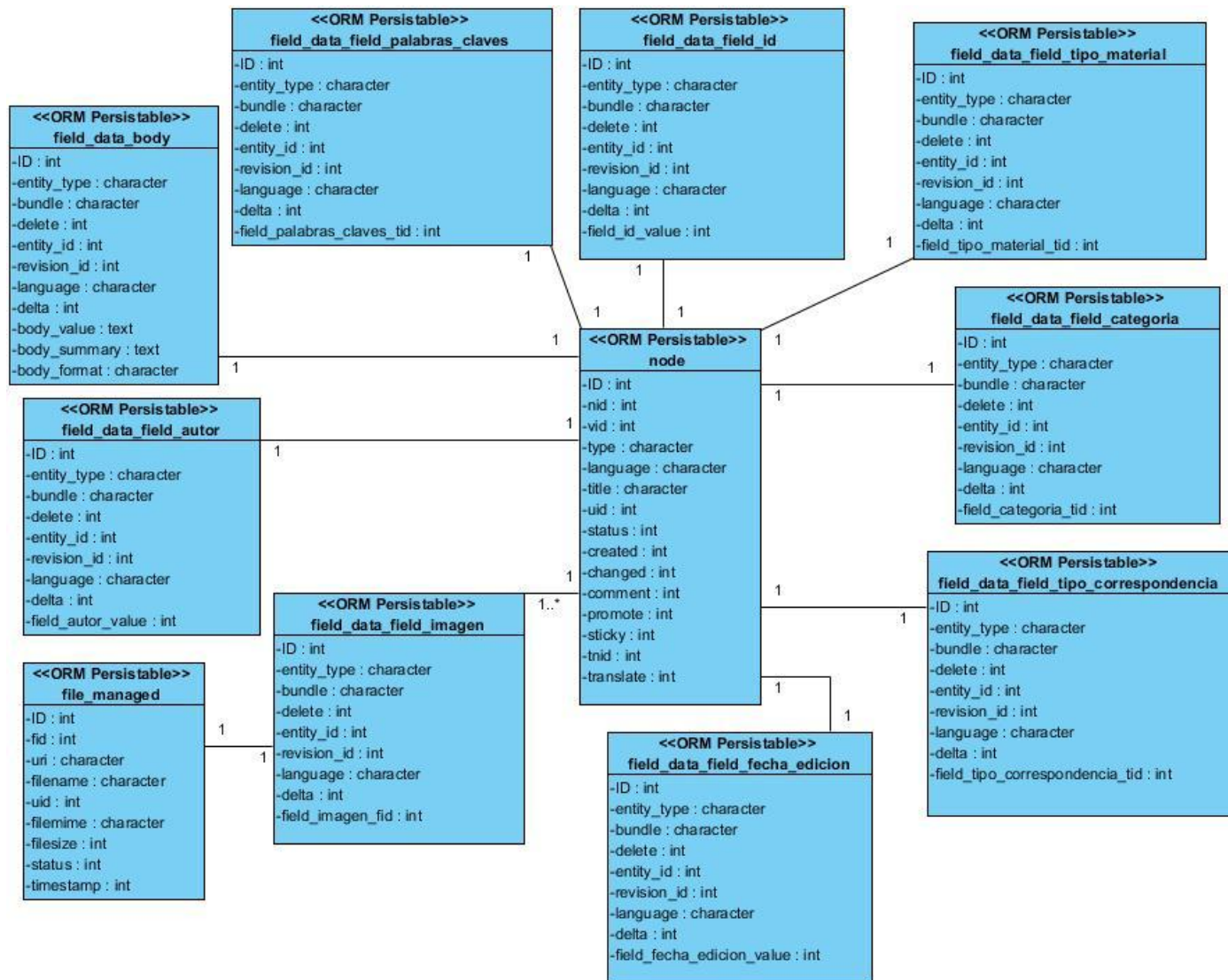


Figura 11. Modelo de clases persistentes.

2.7.2.2 Diagrama Entidad Relación

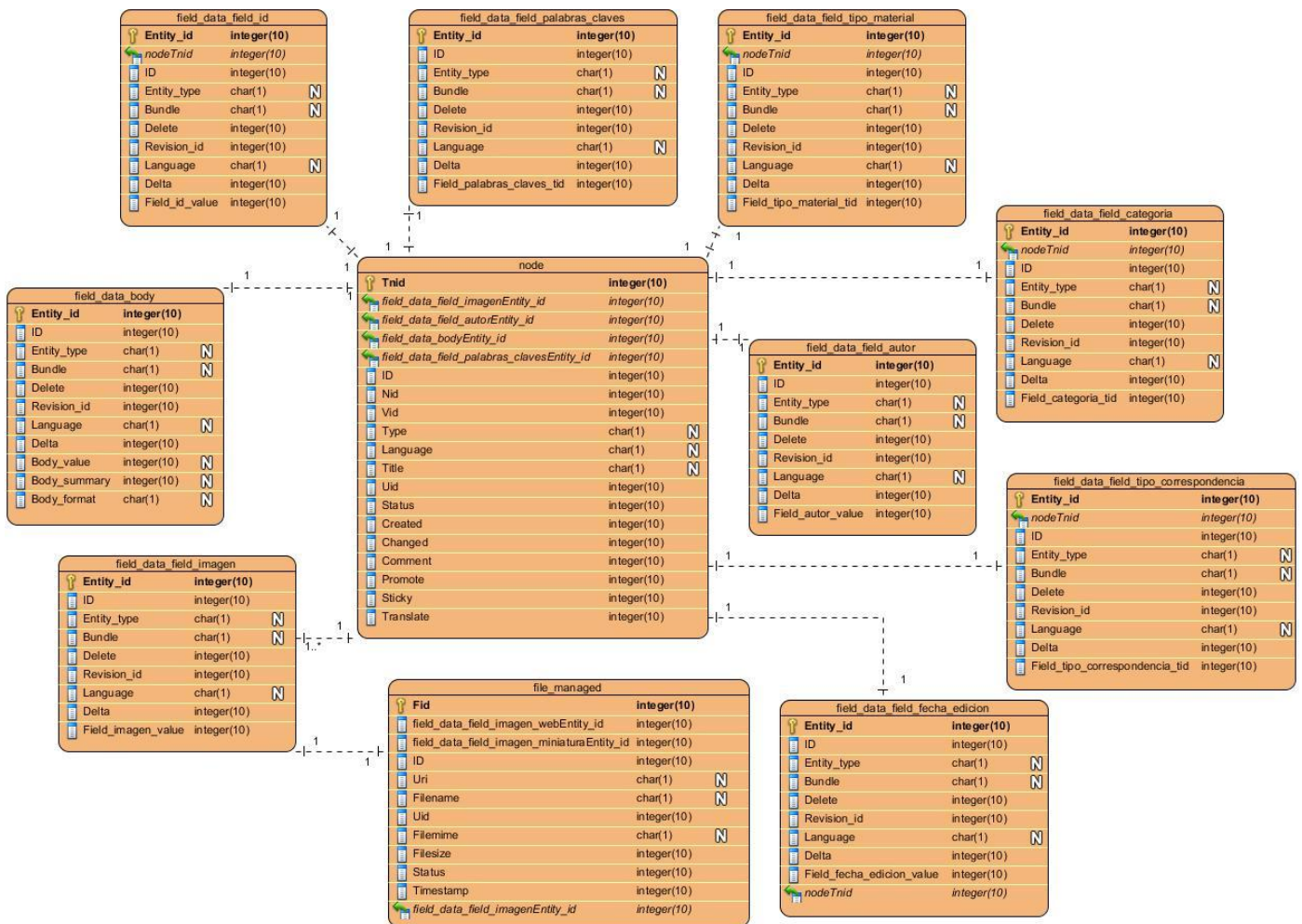


Figura 12. Diagrama Entidad Relación.

Descripción de las tablas.

Las tablas se describen en el [Anexo 6](#).

2.8 Implementación

En el epígrafe anterior se modelaron los diferentes artefactos necesarios para un mejor entendimiento de las funcionalidades del sistema a desarrollar, estos elementos son imprescindibles y sirven de base para la etapa de implementación de la plataforma, para así ubicar al desarrollador a la hora de implementar el sistema y desplugarlo.

2.8.1 Diagrama de despliegue

El diagrama de despliegue permite apreciar de forma visual como se encuentran relacionados físicamente los componentes de la aplicación. El sistema propuesto comprende cuatro elementos descritos a continuación:

PC Cliente: Es la computadora desde donde los usuarios podrán acceder a la Plataforma. Esta PC se conecta vía HTTP con el Servidor Web.

Servidor Web: Es el nodo que realiza las funciones de intermediario entre las PC Clientes que realizan sus peticiones y el servidor donde se encuentra la información. Este servidor web toma los datos, realiza sus funciones y presenta la información a las PC Clientes. A la hora de enviar los correos se conecta al servidor de correo mediante el protocolo TLS.

Servidor de BD: Es el nodo que contiene toda la información del sistema.

Servidor de correo: Es el servidor desde donde se envían los correos.

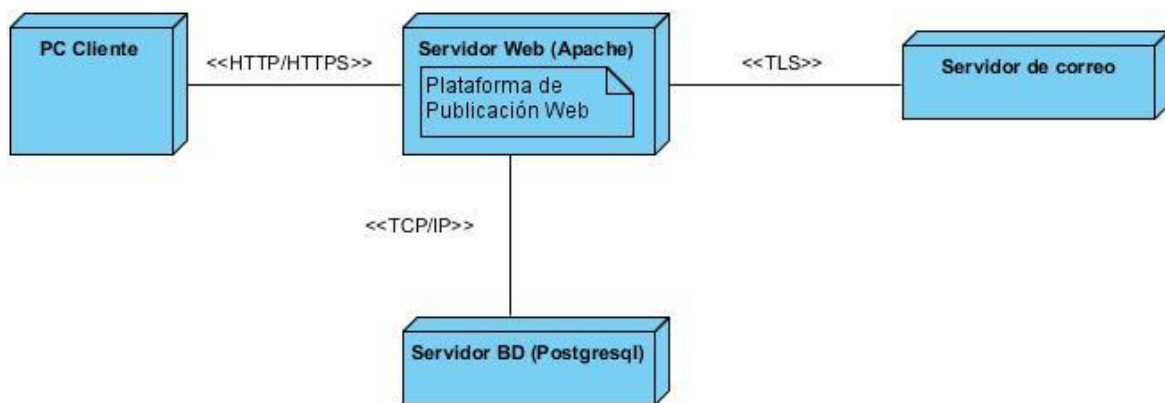


Figura 13. Diagrama de Despliegue.

- HTTP: Es el protocolo usado en cada transacción de la web, define la sintaxis y la semántica que utilizan los elementos *software* de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un URL. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

- TCP/IP: es un conjunto de protocolos. La sigla TCP/IP significa “Protocolo de control de transmisión/Protocolo de Internet”. Proviene del nombre de dos protocolos importantes del conjunto de protocolos, es decir, del protocolo TCP y del protocolo IP. En algunos aspectos, TCP/IP representa todas las reglas de comunicación de Internet y se basa en la noción de dirección IP, es decir, en la idea de brindar una dirección IP a cada equipo de la red para poder enrutar paquetes de datos. Está diseñado para cumplir con una cierta cantidad de criterios, entre ellos: dividir mensajes en paquetes, usar un sistema de direcciones, enrutar datos por la red y detectar errores en las transmisiones de datos.
- TLS (*Transport Layer Security*) es el protocolo que representa una evolución del protocolo SSL (Secure Sockets Layer), es un protocolo mediante el cual se establece una conexión segura por medio de un canal cifrado entre el cliente y servidor, en este caso, entre la Plataforma de Publicación Web y el servidor de correo, lo que permite que el intercambio de información se realice en un entorno seguro y libre de ataques. El uso de estos protocolos permite la prevención de escuchas (eavesdropping), evitan que la identidad del remitente sea falsificada y mantienen la integridad del mensaje en una aplicación cliente-servidor.

2.8.2 Diagrama de componentes

Un diagrama de componentes permite apreciar de forma visual la organización y las dependencias lógicas entre componentes *software*, sean éstos componentes de código fuente, binarios o ejecutables.

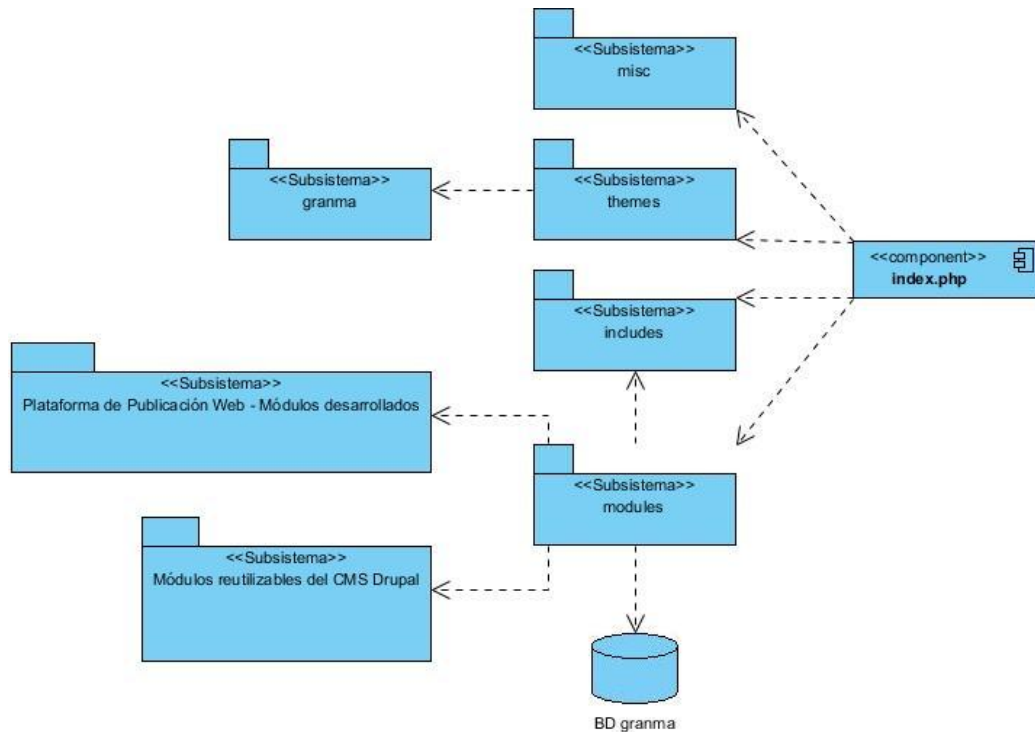


Figura 14. Diagrama de Componentes general.

A continuación una tabla que describe cada uno de los componentes representado en el diagrama anterior:

Componente	Propósito
index.php	Es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS.
themes	Este componente incluye las plantillas que se pueden utilizar, permite separar el contenido de la presentación.
granma	Es la plantilla que contiene un conjunto de ficheros e imágenes para la presentación de la Plataforma.
modules	Este componente contiene todos los módulos, que permiten las distintas funcionalidades del CMS.
includes	Este componente contiene un conjunto de ficheros indispensables para el funcionamiento de Drupal.
misc	Incluye elementos que tienen que ver con el diseño, y funcionamiento (imágenes, ficheros js, etc).
Módulos reutilizables del CMS Drupal	Contiene un conjunto de módulos del CMS Drupal que se utilizan en la Plataforma.
Plataforma de Publicación Web -	Contiene los módulos desarrollados para la Plataforma y que se muestran en la Figura 15.

Módulos desarrollados	
BD granma	Representa la base de datos.

Tabla 4. Descripción de Componentes.

A continuación se muestra el subsistema que contiene los módulos desarrollados para la Plataforma.

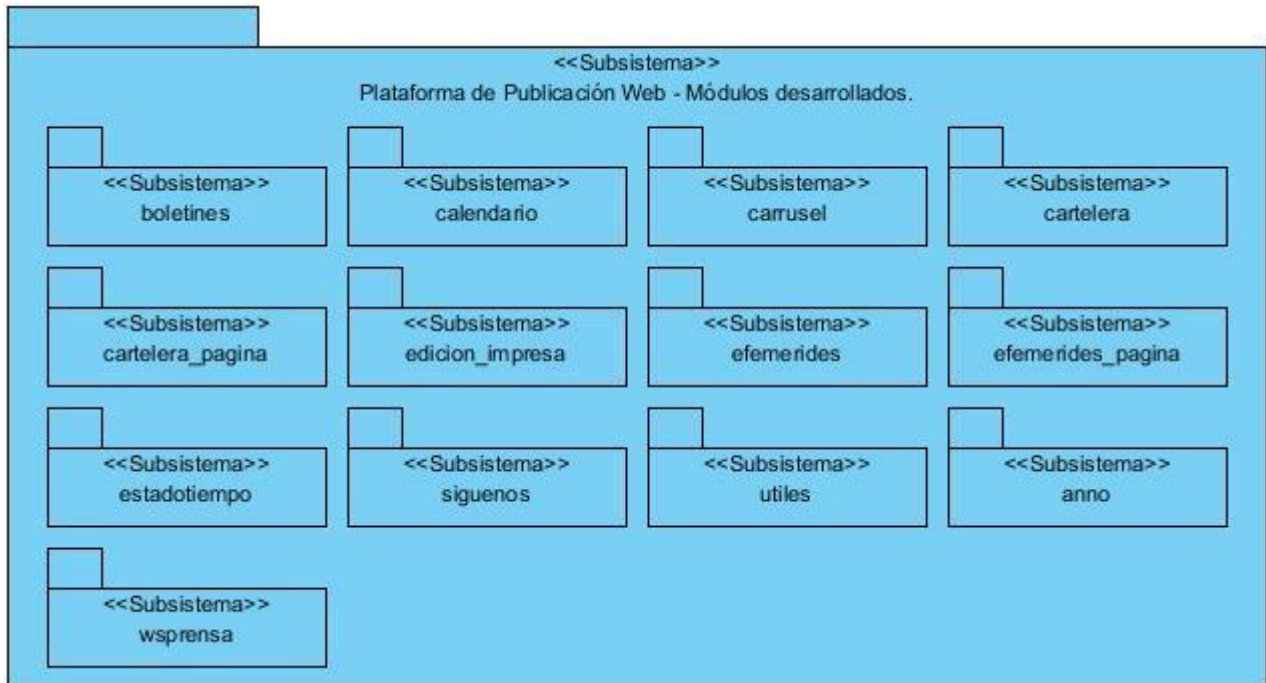


Figura 15. Subsistema Plataforma de Publicación Web - Módulos desarrollados.

A continuación se representa el diagrama de componentes del subsistema (módulo) referente al requisito funcional Mostrar cartelera (RF24) y que pertenece al paquete de módulos desarrollados. Los restantes diagramas se encuentran en el [Anexo 7](#).

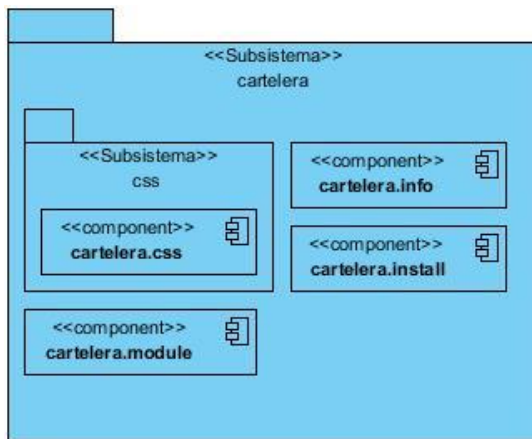


Figura 16. Diagrama de Componentes del subsistema cartelera.

Conclusiones parciales

Se realizó el levantamiento de requisitos que guiaron a desarrollar las distintas funcionalidades de la Plataforma. Se modeló la arquitectura general del sistema a desarrollar, la arquitectura del CMS Drupal y la arquitectura del servicio web a implementar. Se crearon los diferentes artefactos para una correcta implementación de los módulos, respetando siempre las pautas y características del CMS a usar. Se analizaron las características del sistema a desarrollar para modelar un diagrama de despliegue que guiaría el despliegue de la aplicación.

Introducción

Para desarrollar un sistema con calidad es necesario llevar a cabo un plan de pruebas desde el inicio, así como también un desarrollo iterativo y dar un seguimiento a los cambios en el sistema. En el presente capítulo se describen los casos de prueba a los que fueron sometidas las funcionalidades del sistema en desarrollo en cada una de las iteraciones, el cumplimiento de estos casos de pruebas fue el hito para avanzar a la próxima iteración.

3.1 Prueba de software

Las pruebas de *software* son el único instrumento adecuado para determinar el estado de la calidad de un producto de *software*. Se desarrolla un proceso donde se ejecutan pruebas dirigidas a componentes del *software* o al sistema de *software* en su totalidad, con el objetivo de medir el grado en que el *software* cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de prueba.

Las técnicas de pruebas más usadas son las de caja blanca y caja negra, para desarrollar las pruebas de caja blanca es necesario conocer el código del *software* desarrollado, para conocer la lógica interna del sistema a probar y las pruebas de caja negra solo basta con conocer las entradas y salidas del sistema.

A la plataforma se le realizarán pruebas de caja negra para comprobar que las funcionalidades del sistema fueron correctamente implementadas, donde las historias de usuario desempeñarán un papel importante, a cada historia de usuario se le pueden efectuar más de una prueba de funcionalidad, tantas como sean necesarias.

3.2 Casos de prueba

En el desarrollo de una aplicación es necesario entre iteración e iteración desarrollar un conjunto de casos de prueba para poder avanzar a una próxima iteración.

En el desarrollo de la Plataforma de Publicación Web se diseñaron un conjunto de casos de prueba a las que fueron sometidas las diferentes funcionalidades del sistema para comprobar que cumplían de acuerdo con las Historias de usuario. A continuación se describen los escenarios para una de las historias de usuario con prioridad Muy Alta así como los casos de prueba.

Descripción general: La historia de usuario comienza cuando el administrador desea insertar, modificar o eliminar un material periodístico. Se definen qué secciones

Condiciones de ejecución: El administrador debe estar autenticado.

Escenario	EC 1.1 Insertar material periodístico correctamente.		Escenario	EC 1.2 Insertar material periodístico incorrectamente.	
Descripción	El administrador inserta los datos del material periodístico correctamente.		Descripción	El administrador inserta los datos del material periodístico incorrectamente.	
Id	V	http://www.granma.cu:20120509:142	Id	V	http://www.granma.cu:20120509:142
Palabras clave	V	científica, desarrollo, profesionales	Palabras clave	V	científica, desarrollo, profesionales
Palabras posicionamiento	V	Cuba, CUJAE, pedagógicas	Palabras posicionamiento	V	Cuba, CUJAE, pedagógicas
Título	V	En pos de la investigación: más conocimientos	Título	I	(vacío)
Autor	V	Alejandro Gómez	Autor	I	(vacío)
Fuente	V	Prensa Latina	Fuente	V	Prensa Latina
Resumen	V	La X Jornada Científica Estudiantil ha constituido espacio para el intercambio del acontecer investigativo en la Universidad.	Resumen	I	(vacío)
Cuerpo de texto	V	La Facultad 4, a propósito de este espacio, brindó la oportunidad, en horas de la mañana de este miércoles 9 de mayo, de contar con la provechosa conferencia “Valoraciones sobre la investigación científica estudiantil”, del Dr. Honoris Causa del Instituto Superior Politécnico José Antonio Echeverría (CUJAE) y Dr. en Ciencias Pedagógicas, Fernando Vecino Alegret.	Cuerpo de texto	V	La Facultad 4, a propósito de este espacio, brindó la oportunidad, en horas de la mañana de este miércoles 9 de mayo, de contar con la provechosa conferencia “Valoraciones sobre la investigación científica estudiantil”, del Dr. Honoris Causa del Instituto Superior Politécnico José Antonio Echeverría (CUJAE) y Dr. en Ciencias Pedagógicas, Fernando Vecino Alegret.
Imágenes	V	Imágenes adjuntadas	Imágenes	I	(vacío)
Especiales	V	Reflexiones de Fidel	Especiales	V	Reflexiones de Fidel
Categoría	V	nacionales	Categoría	I	(vacío)
Tipo de material	V	Artículo	Tipo de material	V	Artículo

Respuesta del sistema	El sistema verifica que los datos estén correctos, de ser así, almacenará el material periodístico en el sistema.		Respuesta del sistema	El sistema verifica que los datos estén correctos, de no ser así, emitirá varios mensajes de error, uno para cada campo vacío (si estos necesitan ser llenados) indicando que los datos están incorrectos.	
Flujo central	El usuario con los permisos pertinentes accede al menú de administración y luego a Contenido->Agregar contenido y selecciona Material periodístico. El sistema muestra una interfaz para introducir los datos. El usuario introduce los datos y presiona el botón "Guardar".				
Escenario	EC 2.1 Editar material periodístico correctamente.		Escenario	EC 2.2 Editar material periodístico incorrectamente.	
Descripción	El administrador edita los datos del material periodístico correctamente.		Descripción	El administrador edita los datos del material periodístico incorrectamente.	
Id	V	http://www.granma.cu:20120509:142	Id	V	http://www.granma.cu:20120509:142
Palabras clave	V	científica, desarrollo, profesionales	Palabras clave	V	científica, desarrollo, profesionales
Palabras posicionamiento	V	Cuba, CUJAE, pedagógicas	Palabras posicionamiento	V	Cuba, CUJAE, pedagógicas
Título	V	En pos de la investigación: más conocimientos	Título	I	(vacío)
Autor	V	Alejandro Gómez	Autor	I	(vacío)
Fuente	V	Prensa Latina	Fuente	V	Prensa Latina
Resumen	V	La X Jornada Científica Estudiantil ha constituido espacio para el intercambio del acontecer investigativo en la Universidad.	Resumen	I	(vacío)
Cuerpo de texto	V	La Facultad 4, a propósito de este espacio, brindó la oportunidad, en horas de la mañana de este miércoles 9 de mayo, de contar con la provechosa conferencia "Valoraciones sobre la investigación científica estudiantil", del Dr. Honoris Causa del Instituto Superior Politécnico José Antonio Echeverría (CUJAE) y Dr. en Ciencias Pedagógicas, Fernando Vecino Alegret, quien fuera Ministro de Educación Superior (MES) de la República de Cuba durante 30 años.	Cuerpo de texto	V	La Facultad 4, a propósito de este espacio, brindó la oportunidad, en horas de la mañana de este miércoles 9 de mayo, de contar con la provechosa conferencia "Valoraciones sobre la investigación científica estudiantil", del Dr. Honoris Causa del Instituto Superior Politécnico José Antonio Echeverría (CUJAE) y Dr. en Ciencias Pedagógicas, Fernando Vecino Alegret, quien fuera Ministro de Educación Superior (MES) de la República de Cuba durante 30 años.
Imágenes	V	Imágenes adjuntadas	Imágenes	I	(vacío)

Especiales	V	Reflexiones de Fidel	Especiales	V	Reflexiones de Fidel
Categoría	V	nacionales	Categoría	I	(vacío)
Tipo de material	V	Artículo	Tipo de material	V	Artículo
Respuesta del sistema	El sistema verifica que los datos estén correctos, de ser así, modificará el material periodístico en el sistema.		Respuesta del sistema	El sistema verifica que los datos estén correctos, de no ser así, emitirá varios mensajes de error, uno para cada campo vacío (si estos necesitan ser llenados) indicando que los datos están incorrectos.	
Flujo central	El usuario con los permisos pertinentes accede al menú de administración y luego accede a Contenido y da clic en el vínculo editar que se muestra al lado del contenido que desea modificar. El sistema muestra una interfaz para modificar los datos. El usuario modifica los datos y presiona el botón "Guardar".				
Escenario	EC 3.1 Eliminar material periodístico.		Escenario	EC 3.2 Cancelar operación.	
Descripción	En la funcionalidad Eliminar material periodístico no existen variables, solo se deben seguir los pasos del flujo central.		Descripción	En la funcionalidad Cancelar operación no existen variables, solo se deben seguir los pasos del flujo central.	
Respuesta del sistema	El sistema muestra un mensaje para confirmar la eliminación del material periodístico: "Esta acción no se puede deshacer". El usuario selecciona la opción "Eliminar" y el sistema eliminará el material periodístico de la base de datos.		Respuesta del sistema	El sistema cancela la operación y muestra el listado de contenidos existentes en el sistema.	
Flujo central	El usuario con los permisos pertinentes accede al menú de administración y luego accede a Contenido y da clic en el vínculo eliminar que se muestra al lado del contenido que desea eliminar. El sistema muestra una interfaz para confirmar la eliminación. El usuario presiona el botón "Eliminar".		Flujo central	El usuario con los permisos pertinentes accede al menú de administración y luego accede a Contenido y da clic en el vínculo eliminar que se muestra al lado del contenido que desea eliminar. El sistema muestra una interfaz para confirmar o cancelar la eliminación. El usuario presiona el botón "Cancelar".	

Tabla 5. Escenarios del caso de prueba Gestionar material periodístico.

Caso de prueba de funcionalidad	
Código Caso de Prueba: HU-1	Nombre Historia de Usuario: Gestionar material periodístico.
Nombre de la persona que realiza la prueba: Eduardo Manuel Macías Sotolongo	
Descripción de la Prueba: Comprobar la inserción de los materiales periodísticos listos a	

publicar y su correcta publicación en la Plataforma de Publicación Web.

Condiciones de Ejecución:

Que el sistema cumpla con todas las funcionalidades.

Qué el usuario tenga permisos para la inserción de materiales periodísticos.

Entrada / Pasos de ejecución:

En el menú de administración acceder a **Agregar contenido**→**Material periodístico**.

Entrar todos los datos referentes al material periodístico para su publicación, como son: el título, las palabras clave, la localización y fecha, el resumen, autor, fuente, selecciona el tipo de material periodístico, su categoría, el especial al que pertenece en caso que así sea, las imágenes relacionadas y el cuerpo de texto.

Al terminar se introducir los datos del material periodístico se debe dar clic en el botón “Guardar”.

Resultado esperado:

Qué la publicación del material periodístico sea rápida, fácil y eficiente.

Obtener un correcto enfoque del material periodístico.

Evaluación de la Prueba: Satisfactoria.

Campos	Tipos de datos	Reglas o Restricciones
Id	texto	<u>Opcional</u> Puede contener cualquier cadena alfa-numérica. Incluye los siguientes caracteres: _\-.!:\,
Palabras clave	Texto	<u>Opcional</u> Se debe especificar las palabras para relacionar con otro material.
Palabras posicionamiento	Texto	<u>Opcional</u> Se debe especificar las palabras para posicionar el contenido en la web.
Título	Texto	<u>Obligatorio</u> Puede contener cualquier cadena alfa-numérica. Incluye los siguientes caracteres: _\-.!:\ ¿? ,
Autor	Texto	<u>Obligatorio</u> Sólo puede contener letras.
Fuente	Texto	<u>Opcional</u> Sólo puede contener letras.

Resumen	Texto área	<u>Obligatorio</u> Puede contener cualquier carácter.
Cuerpo de texto	Texto área	<u>Obligatorio</u> Puede contener cualquier carácter.
Imágenes	Imagen	<u>Opcional</u> Debe subir archivos menores de 3 MB.
Especiales	Texto	<u>Opcional</u> Puede seleccionar un especial existente o especificar uno nuevo.
Categoría	Lista de selección	<u>Obligatorio</u> Debe seleccionar una categoría, a la que pertenezca el material.
Tipo de material	Lista de selección	<u>Opcional</u> Puede seleccionar un tipo de material.

Tabla 6. Caso de prueba de funcionalidad del Gestionar material periodístico.

Caso de prueba de funcionalidad	
Código Caso de Prueba: HU-1	Nombre Historia de Usuario: Gestionar material periodístico.
Nombre de la persona que realiza la prueba: Eduardo Manuel Macías Sotolongo	
Descripción de la Prueba: Comprobar la modificación de los materiales periodísticos listos a publicar.	
Condiciones de Ejecución: Que el sistema cumpla con todas las funcionalidades. Qué el usuario tenga permisos para la modificación de materiales periodísticos.	
Entrada / Pasos de ejecución: En el menú de administración acceder a Contenido . Buscar el contenido a modificar y dar clic en el vínculo editar . Modificar los datos necesarios del material periodístico. Al terminar se modificar los datos del material periodístico se debe dar clic en el botón “Guardar”.	
Resultado esperado:	

Qué los datos se modifiquen de manera correcta y fácil.

Evaluación de la Prueba: Satisfactoria.

Campos	Tipos de datos	Reglas o Restricciones
Id	texto	<u>Opcional</u> Puede contener cualquier cadena alfa-numérica. Incluye los siguientes caracteres: _\-.!:\,
Palabras clave	Texto	<u>Opcional</u> Se debe especificar las palabras para relacionar con otro material.
Palabras posicionamiento	Texto	<u>Opcional</u> Se debe especificar las palabras para posicionar el contenido en la web.
Título	Texto	<u>Obligatorio</u> Puede contener cualquier cadena alfa-numérica. Incluye los siguientes caracteres: _\-.!:\ &?\'
Autor	Texto	<u>Obligatorio</u> Sólo puede contener letras.
Fuente	Texto	<u>Opcional</u> Sólo puede contener letras.
Resumen	Texto área	<u>Obligatorio</u> Puede contener cualquier carácter.
Cuerpo de texto	Texto área	<u>Obligatorio</u> Puede contener cualquier carácter.
Imágenes	Imagen	<u>Opcional</u> Debe subir archivos menores de 3 MB.
Especiales	Texto	<u>Opcional</u> Puede seleccionar un especial existente o especificar uno nuevo.
Categoría	Lista de selección	<u>Obligatorio</u> Debe seleccionar una categoría, a la que pertenezca el material.
Tipo de material	Lista de selección	<u>Opcional</u> Puede seleccionar un tipo de material.

Tabla 7. Caso de prueba de funcionalidad del Gestionar material periodístico.

Caso de prueba de funcionalidad

Código Caso de Prueba: HU-1	Nombre Historia de Usuario: Gestionar material periodístico.
Nombre de la persona que realiza la prueba: Eduardo Manuel Macías Sotolongo	
Descripción de la Prueba: Comprobar la eliminación de los materiales periodísticos.	
Condiciones de Ejecución: Que el sistema cumpla con todas las funcionalidades. Qué el usuario tenga permisos para la eliminación de materiales periodísticos.	
Entrada / Pasos de ejecución: En el menú de administración acceder a Contenido . Buscar el contenido a eliminar y dar clic en el vínculo eliminar . Confirmar la eliminación en la nueva página que se muestra dando clic en “Eliminar”.	
Resultado esperado: Qué los datos se eliminen de manera correcta.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 8. Caso de prueba de funcionalidad del Gestionar material periodístico.

Los restantes casos de prueba de funcionalidad se encuentran en el [Anexo 8](#).

Los casos de pruebas que se describen a continuación son los que se diseñaron para comprobar que las funcionalidades de integración cumplen correctamente según las historias de usuario relacionadas con la integración entre la Plataforma de Publicación Web, el Sistema de Gestión Documental y el Sistema de Gestión Editorial. A continuación se describen los casos de prueba para una de las historias de usuario con prioridad Muy Alta.

Caso de prueba de integración	
Código Caso de Prueba: CPHU-17.1	Nombre Historia de Usuario: HU-17
Nombre de la persona que realiza la prueba: Ricardo Valdivia Freyre	
Descripción de la Prueba: Prueba de integración entre sistemas para la creación de contenidos	

en Plataforma de Publicación Web para periódico Granma.
Condiciones de Ejecución: Tener habilitado en el sistema el protocolo SOAP para establecer una conexión con servicio web en la Plataforma de Publicación Web para periódico Granma.
<p>Entrada / Pasos de ejecución:</p> <p>Entradas:</p> <ol style="list-style-type: none"> 1. Nombre de usuario en la Plataforma de Publicación Web para periódico Granma para Gestión Editorial. 2. Contraseña de usuario. 3. Nombre del archivo a subir al sistema con los contenidos para publicar. 4. Cuerpo del archivo codificado en base64. <p>Pasos de ejecución</p> <p>Desde un sistema remoto se generan los contenidos en formato XML para Edición Impresa, Cartelera, Efemérides y Estado del tiempo y para Materiales Periodísticos, Correspondencia, Galería de imágenes y Opinión gráfica se genera un XML bajo estándar IPTC. Contenidos en XML y archivos asociados tales como imágenes y PDF son comprimidos en formato “zip”.</p> <p>Paso 1: Establecer comunicación con servicio web SOAP</p> <p>Paso 2: Entrar datos a función de acceso; revisar WSDL, el nombre de función varía dinámicamente según el ENDPOINT establecido en la configuración de módulo Services. El usuario o la contraseña son incorrectos.</p>
<p>Resultado Esperado:</p> <p>Respuesta a) Entra usuario o contraseña incorrecto. Aparece el mensaje: “Debe ser un usuario válido del sistema”.</p> <p>Respuesta b) Intenta entrar anónimamente. Aparece el mensaje: “Debe ser un usuario válido y autenticado en el sistema”.</p>
Evaluación de la Prueba: Satisfactoria

Tabla 9. Caso de prueba de integración del Gestionar material periodístico.

Caso de prueba de integración	
Código Caso de Prueba: CPHU-17.2	Nombre Historia de Usuario: HU-17

Nombre de la persona que realiza la prueba: Ricardo Valdivia Freyre
Descripción de la Prueba: Prueba de integración entre sistemas para la creación de contenidos en Plataforma de Publicación Web para periódico Granma.
Condiciones de Ejecución: Tener habilitado en el sistema el protocolo SOAP para establecer una conexión con servicio web en la Plataforma de Publicación Web para periódico Granma.
<p>Entrada / Pasos de ejecución:</p> <p>Entradas:</p> <ol style="list-style-type: none"> 1. Nombre de usuario en la Plataforma de Publicación Web para periódico Granma para Gestión Editorial. 2. Contraseña de usuario. 3. Nombre del archivo a subir al sistema con los contenidos para publicar. 4. Cuerpo del archivo codificado en base64. <p>Pasos de ejecución</p> <p>Desde un sistema remoto se generan los contenidos en formato XML para Edición Impresa, Cartelera, Efemérides y Estado del tiempo y para Materiales Periodísticos, Correspondencia, Galería de imágenes y Opinión gráfica se genera un XML bajo estándar IPTC. Contenidos en XML y archivos asociados tales como imágenes y PDF son comprimidos en formato “zip”.</p> <p>Paso 1: Establecer comunicación con servicio web SOAP</p> <p>Paso 2: Entrar datos a función de acceso (revisar WSDL, nombre de función varia dinámicamente según el ENDPOINT establecido en la configuración de módulo Services). Carga un fichero distinto al tipo “tar.gz”.</p>
<p>Resultado Esperado:</p> <p>Respuesta a) El archivo subido no es de extensión “zip”. Aparece el mensaje: “No se realizó la transferencia. El fichero debe ser un compactado zip”.</p>
Evaluación de la Prueba: Satisfactoria.

Tabla 10. Caso de prueba de integración del Gestionar material periodístico.

Los restantes casos de prueba de integración se encuentran en el [Anexo 9](#).

Resultado de las pruebas de funcionalidad e integración

Se probaron 46 casos de pruebas que responden a las funcionalidades de la Plataforma, los cuales arrojaron 20 no conformidades que fueron resueltas por el equipo de desarrollo y se probaron además 6 casos de prueba que responden a las funcionalidades de integración que tuvieron un resultado satisfactorio para todo los casos.

3.3 Pruebas de rendimiento.

Las pruebas de rendimiento son necesarias para comprobar que el sistema cumple con los criterios de rendimiento, para comparar dos sistemas para encontrar cuál de ellos funciona mejor o para medir que partes del sistema o de carga de trabajo provocan que rinda de forma negativa. Para el diagnóstico se utilizan diferentes tipos de pruebas y herramientas de *software*. Los tipos de pruebas de rendimiento que se le realizaron a la Plataforma fueron las Pruebas de Carga y Stress y como herramienta JMeter, para esto se tomó una muestra de 200 usuarios conectados concurrentemente, para comprobar que las funcionalidades tengan un buen rendimiento.

3.3.1 Herramienta.

JMeter es un proyecto que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web.

JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas. JMeter puede también ser configurado como un monitor, aunque es comúnmente considerado una solución ad-hoc respecto a soluciones avanzadas de monitoreo. Soporta aserciones para asegurarse que los datos recibidos son correctos, por cookies de hilos, configuración de variables y una variedad de reportes.

3.3.2 Tipos de Pruebas.

Pruebas de Carga.

Las pruebas de carga son el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. La ejecución de estas pruebas con carga genera resultados, incluidos contadores de rendimiento y de otros tipos, en tablas y en gráficos.

Por ejemplo, esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos y el servidor de aplicaciones también se monitorizan, entonces esta prueba puede mostrar el cuello de botella en la aplicación.

Pruebas de Stress.

Las pruebas de stress son pruebas de carga y rendimiento basadas en la funcionalidad del sistema bajo cargas pesadas, un gran número de repeticiones, manejo de grandes datos y demasiadas preguntas a bases de datos grandes. Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

A continuación se diseña el plan de pruebas de rendimiento para las diferentes acciones que los usuarios pueden realizar al conectarse a la Plataforma.

Funcionalidad	URL
Portada	granma/portada
Galerías	granma/galerias
Opiniones	granma/opiniones
Contenido	granma/node/71
Especiales	granma/especiales
Listado de materiales nacionales	granma/categoría/nacionales

Tabla 11. Resultado de las pruebas de rendimiento.

Para una muestra de 200 usuarios conectados concurrentemente, con un período de subida de 1 segundo (tiempo que espera cada usuario para realizar una petición) la aplicación generó un total de 758333.1 kb/s de transferencia de datos y un rendimiento de 87.8/s. Funciona de manera aceptable y estable para 74600

peticiones realizadas por los usuarios en total y responde con una media de tiempo de 1.812 segundos. Para el desarrollo de las pruebas se seleccionaron dos computadoras, cada una con un procesador Intel Core 2 Duo y 2GB de RAM, una para alojar la aplicación y la otra para alojar la base de datos.

3.4 Resultados obtenidos

- Gestión de los contenidos a publicar en la plataforma.
- Gestión de las páginas estáticas encargadas de mostrar información adicional.
- Gestión de las galerías de imágenes y opiniones gráficas.
- Gestión de boletines para enviar desde el periódico los últimos contenidos publicados.
- Búsqueda básica y avanzada de contenidos.
- Acceso a documentos digitales con un resumen de la información de la edición impresa.
- Interacción con la plataforma por medio de encuestas.
- Mostrar información climatológica de las distintas regiones del país.
- Mostrar la cartelera televisiva de los distintos canales de la Televisión cubana.
- Integración con el Sistema de Gestión Editorial y el Sistema de Gestión Documental.
- Gestión de usuarios y roles dentro de la aplicación.
- Sistema de impresión de información.
- Seguimiento de los contenidos más visitados.

Conclusiones parciales

Se describen los casos de prueba de funcionalidad e integración que guiaron la calidad de la aplicación y que fueron la base a seguir desarrollando otras funcionalidades, donde de los cuarenta y seis casos de pruebas de funcionalidad diseñados, luego de solucionar las no conformidades detectadas en las pruebas internas, se obtuvo un resultado satisfactorio en todos y de los seis casos de prueba de integración diseñados se obtuvo en todos un resultado satisfactorio. Se seleccionaron los tipos de pruebas a utilizar para el estudio del rendimiento de la aplicación y se describieron las diferentes características de la herramienta a utilizar en las pruebas de carga. En el análisis de los resultados obtenidos se muestran las funcionalidades con las que cuenta la Plataforma de Publicación Web.

- La investigación realizada permitió identificar los diferentes conceptos y características de los medios de prensa en la actualidad.
- Como resultado de la investigación se decidió desarrollar una Plataforma de Publicación Web que posibilitara la publicación de los contenidos que se gestionan en la redacción del periódico Granma.
- Como propone la metodología seleccionada se generaron los artefactos necesarios para la modelación del sistema.
- Para comprobar la calidad de la aplicación se diseñaron y se efectuaron casos de prueba que fueron la base para verificar el cumplimiento de los requisitos.

Recomendaciones

- Desarrollar funcionalidades que permitan la personalización rápida de los contenidos a publicar y de esta manera facilitar la usabilidad de la información en la Plataforma.
- Capacitación a los usuarios que van a administrar la Plataforma para un mejor entendimiento de las funcionalidades con las que cuenta el sistema.
- Trazar una política de actualizaciones periódicas utilizando el módulo de actualización del CMS Drupal, con el objetivo de evitar fallas de seguridad o mal funcionamiento del sistema.

Referencias Bibliográficas

- Aliaga, Antonio, y Marcos Agustín Miani. *PostgreSQL*. enero 2011 Available from world wide web: <<http://www.iessanvicente.com/colaboraciones/postgreSQL.pdf>>.
- Álvarez, Miguel Ángel. Qué es PHP. *desarrolloweb.com* 2001. [citado 8 noviembre 2011]. Available from world wide web: <<http://www.desarrolloweb.com/articulos/392.php>>.
- Álvarez, Sara. Sistemas gestores de bases de datos. *desarrolloweb.com* 2007. [citado 22 septiembre 2011]. Available from world wide web: <<http://www.desarrolloWeb.com/articulos/sistemas-gestores-bases-datos.html>>.
- Anón. World Wide Web. *World Wide Web* 2008. [citado 16 octubre 2011]. Available from world wide web: <<http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>>.
- Bradenbaugh, Jerry. *Aplicaciones JavaScript*. 2000 [citado 29 noviembre 2011]. Available from world wide web: .
- Briones, K., Maliza, C., y Ochoa, X. *Análisis, diseño e implementación de una aplicación multimedia interactiva para mostrar tiempos, distancias y rutas en un sistema de transporte masivo urbano utilizando herramientas de software libre y tecnología de web 2.0*. agosto 2009.
- Eloi de San Martín. Programación Web. *ProgramacionWeb.net* febrero 2005. [citado 13 octubre 2011]. Available from world wide web: <<http://www.programacionweb.net/articulos/articulo/?num=287>>.
- Máster P. Intelectual. Curso de Ajax. *ProgramacionWeb.net* 2005. [citado 18 noviembre 2011]. Available from world wide web: <<http://www.programacionweb.net/cursos/curso.php?num=2>>.
- Mercer, David. *Building powerful and robust websites with Drupal 6*. abril 2008 [citado 20 noviembre 2011]. Available from world wide web: .
- Parra, Eduardo. Instalar Netbeans 7.0 en español en Ubuntu. abril 2011. [citado 2 octubre 2011]. Available from world wide web: <<http://www.portalubuntu.com/2011/04/instalar-netbeans-70-en-espanol-en.html>>.
- Pérez, Javier Eguíluz. *Introducción a AJAX*. junio 2008 [citado 26 octubre 2011]. Available from world wide web: .
- Pérez, Javier Eguíluz. *Introducción a JavaScript*. [citado 16 noviembre 2011]. Available from world wide web: .
- Pérez Valdés, Damián. Framework de PHP. *Maestros del Web* 2007. [citado 1 diciembre 2011]. Available from world wide web: <<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>>.

Piattini. *INGENIERIA DE SOFTWARE EDUCATIVO*. 1996.

Safuto, Robert J. *Beginners Guide to Drupal*. 2011 [citado 7 octubre 2011]. Available from world wide web:

Sierra, María. *Trabajando con Visual Paradigm for UML*. 2008. Available from world wide web: <<http://personales.unican.es/ruizfr/is1/doc/lab/01/is1-p01-trans.pdf>>.

The Apache Software Foundation. *The Apache Software Foundation*. 2011. Available from world wide web: <http://httpd.apache.org/docs/2.0/es/new_features_2_0.html>.

Tomlinson, Todd, John VanDyk, y Matt Westgate. *Pro Drupal Development*. 2007 [citado 17 noviembre 2011]. Available from world wide web: .

Vargas, Guillermo de Jesús Saldivar. *Diseño de bases de datos*. 2005. [citado 16 febrero 2012]. Available from world wide web: <<http://www.monografias.com/trabajos30/base-datos/base-datos.shtml>>.

Drupal Association. 2010. Inglés. <http://drupal.org/>

Gladys Marsi Peñalver Romero. *Metodología ágil para proyectos de software libre.* 2008. Español.

Alejandro Rost. *El Periódico Digital: características de un nuevo medio.* 2003. Español. <http://red-accion.uncoma.edu.ar/assignaturas/pd.htm>

Periódico Granma. *Diario Granma.* 2011. <http://www.granma.cubaweb.cu>

Lissabet, Gilberto. *Proyecto Técnico Gestión Documental para la Prensa.* 2011.

Aparicio, Yankiel. *Proyecto Técnico Gestión Editorial para la Prensa.* 2011.

Juglar. *Sistema de Gestión de Contenidos.* 2007. <http://juglar103.blogspot.com/2004/09/01/sistemas-de-gestion-de-contenidos/>

García, Xavier. *Introducción a los Sistemas de Gestión de Contenidos de código abierto.* 2004. <http://mosaic.uoc.edu/articulos/cms1204.html>

Martin Gersbach, José A. Reyero y Jairo Enrique Serrano Castañeda. *H2O Drupal Hispano - Comunidad de usuarios de Drupal.* 2010. <http://drupal.org.es/>

GoldScripts. *Soluciones inteligentes para webmasters exitosos.* 2007. <http://www.goldscripsts.com/2007/11/14/wordpress-caracteristicas-generales/>

Upton, David. *Codeigniter for rapid php application development.* 2007

Menéndez Guerra, Frank Orlando. *Migración del subsistema gestión de Cuadros y Personal de Apoyo del proyecto de Gestión Fiscal a la versión 1.3 del Framework Symfony.* 2010. http://bibliodoc.uci.cu/TD/TD_03309_10.pdf

Thomas Lockhart, Equipo de desarrollo de PostgreSQL. *Tutorial de PostgreSQL.* 1999. <http://lib.undersecurity.net/index.php?download=../Postgres-Tutorial.pdf>

David Axmark y Michael Widenius. *MySQL 5.0 Manual - Introducción.* 2001. <http://dev.mysql.com/doc/refman/5.0/es/tutorial.html>

ABCdatos. 2011. Servidor Apache v.2.2.21 <http://www.abcdatos.com/webmasters/programa/z2818.html>

Rafael Camps Paré. 2001. *Cliente gráfico: pgAdmin3.* <http://www.dataprix.com/8-cliente-grafico-pgadmin3>

W3C. 2008. *World Wide web.* <http://www.w3c.es/divulgacion/guiasbreves/XHTML>

- Miguel Ángel Álvarez.** 2010. *Manual de JQuery.*
<http://sunshine.prod.uci.cu/gridfs/sunshine/books/manual-jquery.pdf>
- Córdova Díaz.** 2009. *Sistema para la gestión de los proceso de transporte aéreo de carga.*
<http://bibdigital.epn.edu.ec/bitstream/15000/1556/1/CD-2705.pdf>
- Jennifer Hodgdon.** 2009. *Drupal Association.* Inglés. <http://drupal.org/node/547518>
- W3C Consortium.** *Web Services Architecture.* 2004. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>
- XML-RPC.com.** <http://xmlrpc.scripting.com/default.html>
- W3C SOAP.** *SOAP Specifications.* 2007. <http://www.w3.org/TR/soap/>
- Roy Thomas Fielding.** *Architectural Styles and the Design of Network-based Software Architectures.* 2000. <http://roy.gbiv.com/pubs/dissertation/top.htm>
- Sourceforge.** NuSOAP - SOAP Toolkit for PHP | Get NuSOAP - SOAP Toolkit for PHP at SourceForge.net. 2010 <http://sourceforge.net/projects/nusoup/>
- Programer's Reference Guide.* 2008. <http://www.zend.com>
- Marcos Andrés Bonvin Faura.** LA PRENSA DIGITAL: LENGUAJE Y CARACTERÍSTICAS. 2007. <http://hera.ugr.es/tesisugr/17116880.pdf>
- Pruebas de software.** *Prueba de software.*2005. <http://pruebasdesoftware.com/laspruebasdesoftware.htm>
- Diovis Robinet y Yerandi Bracero.** *Sistema de prueba de rendimiento para SoftPBX Asterisk.* 2009. http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_2703_09/1/TD_2703_09.pdf
- Mariela Valdés Fernández.** *Propuesta de un procedimiento para la evaluación del rendimiento de las bases de datos basadas en PostgreSQL para software de gestión.* 2010. http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_02937_10/1/TD_02937_10.pdf

Arquitectura: *Arquitectura del Software* o *Arquitectura Lógica*. Conjunto de elementos de programación adecuadamente estructurados dentro de un sistema, con el fin de crear una programación lógica y fiable para el diseño de aplicaciones.

Cliente-Servidor: *Arquitectura Cliente-Servidor*. Protocolo que permite que una aplicación solicite una información o un suceso a otra aplicación (normalmente ajena) para que ésta, realice la tarea adecuada. Se suele utilizar en Internet.

Internet: Internet es la red de redes donde reside toda la información. Es un conjunto de redes, redes de ordenadores y equipos físicamente unidos mediante cables que conectan puntos de todo el mundo.

Patrón de arquitectura: Es una solución a un problema, que se puede aplicar muchas veces, en distintas situaciones.

Plugins: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Rol: Conjunto de acciones que realiza una persona como actor.

Script: Es un guión o conjunto de instrucciones.

Servidor: Máquina que responde a las peticiones de los clientes con el fin de servirles los servicios o recursos que están solicitando. Sigue la arquitectura Cliente-Servidor.

Software: Es un término que viene a decir "*Parte Lógica*" y es el conjunto de elementos lógicos que componen un sistema informático.

SQL: *Structured Query Language: Lenguaje Estructurado de Petición*. Es un lenguaje de programación, especializado para realizar consultas a bases de datos. Hoy en día, se puede considerar un estándar para la realización de consultas en multitud de SGBD.

URL: Acrónimo de *Uniform Resource Locator* (Localizador de Recursos Uniforme), dirección de una página web.

Widget: 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.

W3C: *World Wide Web Consortium*. Comunidad internacional que desarrolla estándares que aseguran el crecimiento de la web a largo plazo, el objetivo principal es guiar la web hacia su máximo potencial.

Web: Sistema lógico de acceso y búsqueda de la información disponible en Internet, cuyas unidades informativas son las páginas web, páginas a las que se puede acceder usando un navegador.

Web 2.0: Término usado para identificar aquellas páginas cuya tecnología supera las páginas web que sean estáticas. Suelen ser webs que contienen acceso a bases de datos.

XML: Metalenguaje capaz de definir sus propias etiquetas.

Anexo 1. Lista de Reserva del Producto.

La Lista de Reserva del Producto (LRP) está conformada por una lista priorizada que define el trabajo a realizar en el proyecto. El objetivo es asegurar que el producto definido al terminar la lista, sea el más correcto, útil y competitivo posible. Solo puede ser modificado entre iteraciones.

Prioridad	Elemento	Descripción	Estimación	Estimado por
Muy Alta				
	1	Insertar material periodístico	0.8	Programador
	2	Modificar material periodístico	0.8	Programador
	3	Eliminar material periodístico	0.8	Programador
	4	Mostrar material periodístico	0.8	Programador
	5	Insertar páginas	0.3	Programador
	6	Modificar páginas	0.3	Programador
	7	Eliminar páginas	0.3	Programador
	8	Mostrar páginas	0.3	Programador
	9	Insertar correspondencia	0.4	Programador
	10	Modificar correspondencia	0.4	Programador
	11	Eliminar correspondencia	0.4	Programador
	12	Mostrar correspondencia	0.4	Programador
	13	Insertar estado del tiempo	1	Programador
	14	Modificar estado del tiempo	1	Programador
	15	Eliminar estado del tiempo	1	Programador
	16	Mostrar estado del tiempo	1	Programador
	17	Insertar pancartas publicitarias	0.4	Programador
	18	Modificar pancartas publicitarias	0.4	Programador
	19	Eliminar pancartas publicitarias	0.4	Programador
	20	Mostrar pancartas publicitarias	0.4	Programador
	21	Insertar cartelera	1	Programador
	22	Modificar cartelera	1	Programador
	23	Eliminar cartelera	1	Programador
	24	Mostrar cartelera	1	Programador
	25	Insertar efemérides	1	Programador
	26	Modificar efemérides	1	Programador
	27	Eliminar efemérides	1	Programador
	28	Mostrar efemérides	1	Programador
	29	Insertar boletín	1	Programador
	30	Eliminar boletín	1	Programador
	31	Suscribir al boletín	1	Programador

	32	Dar de baja del boletín	1	Programador
	33	Insertar edición impresa	1	Programador
	34	Modificar edición impresa	1	Programador
	35	Eliminar edición impresa	1	Programador
	36	Mostrar edición impresa	1	Programador
	37	Insertar galería de imágenes	0.7	Programador
	38	Modificar galería de imágenes	0.7	Programador
	39	Eliminar galería de imágenes	0.7	Programador
	40	Mostrar galería de imágenes	0.7	Programador
	41	Insertar opinión gráfica	0.7	Programador
	42	Modificar opinión gráfica	0.7	Programador
	43	Eliminar opinión gráfica	0.7	Programador
	44	Mostrar opinión gráfica	0.7	Programador
	45	Insertar encuesta	0.3	Programador
	46	Modificar encuesta	0.3	Programador
	47	Eliminar encuesta	0.3	Programador
	48	Mostrar encuesta	0.3	Programador
	49	Insertar enlaces	0.3	Programador
	50	Modificar enlaces	0.3	Programador
	51	Eliminar enlaces	0.3	Programador
	52	Mostrar enlaces	0.3	Programador
	53	Insertar síguenos en	0.3	Programador
	54	Modificar síguenos en	0.3	Programador
	55	Eliminar síguenos en	0.3	Programador
	56	Mostrar síguenos en	0.3	Programador
	57	Insertar especial	0.3	Programador
	58	Modificar especial	0.3	Programador
	59	Eliminar especial	0.3	Programador
	60	Mostrar especial	0.3	Programador
	61	Brindar servicio web para la inserción de contenidos	2	Programador
	62	Insertar contenido mediante archivo	1	Programador
	63	Mostrar listado de contenidos periodísticos por categoría	1	Programador
Alta				
	64	Mostrar lo más visto	0.5	Programador
	65	Mostrar calendario	1	Programador
Media				
	66	Mostrar artículos relacionados	1	Programador
Baja				
	67	Mostrar conozca a Cuba	0.3	Programador
	68	Mostrar quiénes Somos	0.3	Programador
RNF				

Usabilidad	69	El sistema debe permitir un alto nivel de facilidades de uso, basado en el cumplimiento de los siguientes aspectos:	1	Programador
		Los usuarios que accedan a la Plataforma pueden tener sexo femenino o masculino, de cualquier edad, nivel educacional, ocupación, experiencia profesional y puede no tener alguna experiencia con la aplicación.		
		El tiempo que pueden llevar los usuarios anónimos en adaptarse a la aplicación es de un día y para el usuario administrador dos días.		
		Los menús permitirán una navegación sencilla, tanto a los usuarios con conocimientos avanzados de informática como a los usuarios más inexpertos.		
		Se proporcionará a los usuarios el manual de usuario y toda la documentación necesaria para lograr un entendimiento total de las acciones a realizar en el sistema.		
		El sistema debe presentar las siguientes características:		
		Ambiente <ul style="list-style-type: none"> • Servidor web: Apache en su versión 2.0. • Lenguaje de programación: PHP en su versión 5.3. • El intérprete de aplicaciones web: Navegador. • Sistema gestor de bases de datos: Postgres. 		
		Hardware mínimo <ul style="list-style-type: none"> • Un servidor de gama media a alta. • 8GB de memoria RAM. • Discos duros rápidos en el orden de los 10KRPM. • Al menos dos procesadores de doble núcleo, que tengan fuente redundante. • Tarjeta de red a 1GB Ethernet. 		
		Hardware ideal <ul style="list-style-type: none"> • 2 servidores con las 		

		características anteriormente mencionadas, uno para alojar al Drupal con el apache y otro para alojar la base datos con PostgreSQL.		
Interfaz	70	El sistema debe ser compatible con los navegadores Mozilla Firefox, Internet Explorer, Safari, Opera y Chrome.		
Portabilidad	71	El sistema debe ser multiplataforma, principalmente compatible con los sistemas operativos más usados, como son: Windows y GNU/Linux.		
Seguridad	72	Ante cualquier falla en el sistema, se deben mostrar los errores sin dar detalles de información, que pudiera comprometer la seguridad e integridad del mismo.		
		El sistema debe estar disponible el 98% del tiempo, donde 2% restante se debe dedicar al soporte, actualización y corrección de errores.		
		Debe contarse con un sistema de salvadas externas de la información, para casos de desastres.		
Eficiencia	73	Establecer como línea base la rapidez y eficiencia de las operaciones, tanto de acceso a datos como al mostrar la información.		
		La plataforma debe presentar un bajo nivel de carga para asegurar un tiempo de respuesta rápido a las peticiones de los lectores.		
Soporte	74	El sistema debe dar la posibilidad de ser mejorado y de incorporarle nuevos servicios en caso de ser necesarios.		

Tabla 12. Lista de Reserva del Producto.

Anexo 2. Historias de usuario.

Se especifica cada una por separado con las tareas ingenieriles en los documentos contenidos en la carpeta “Historias de usuario”, adjuntada al documento del trabajo de diploma.

Anexo 3. Lista de Riesgos.

No	Riesgo	Tipos de	Impacto	Descripción	Probabilidad	Efectos	Mitigación del
----	--------	----------	---------	-------------	--------------	---------	----------------

		riesgos					riesgo
1	Alcance u objetivos no claros o mal entendidos.	Organización		Los objetivos del proyecto no están lo suficientemente claros para todo el personal de desarrollo.	Media	Serios	
2	Abandono de un miembro del proyecto.	Personal		Algún miembro del equipo de desarrollo se retira por cualquier razón.	Baja	Serios	
3	Apagón en local.	Tecnológico		Ocurre un apagón, se está desarrollando el sitio y se apagan las máquinas del local repentinamente.	Media	Tolerable	Realizar copias de respaldo cada día y cada un tiempo regular para evitar perdida o afectación en los datos.
4	Bajo conocimiento del lenguaje de Programación	Personal		Algún miembro del grupo de desarrollo no tiene los conocimientos necesarios del lenguaje de Programación con el que se va a implementar el proyecto.	Baja	Tolerable	Realizar capacitaciones o talleres para crear conocimientos básicos sobre el tema.
5	Problemas familiares	Personal		No se cuenta con el personal calificado para realizar una determinada tarea por encontrarse fuera del proyecto por un tiempo indeterminado.	Media	Tolerable	Buscar personal que pueda ocupar de manera temporal o permanente el lugar en el proceso de desarrollo.
6	Falta de compromiso por parte de la alta dirección.	Personal		Los dirigentes no se sienten comprometidos con el proyecto, y por tanto no le prestan la atención requerida, lo que lleva a la toma de	Baja	Tolerable	Mantener comunicación constante entre las partes asociadas, de acuerdo a los compromisos acordados para la

				decisiones inadecuadas.			selección del personal.
7	Irregularidades en el control de los cambios	Organización		Los cambios que se realizan por parte de desarrolladores, analistas y otros involucrados en el desarrollo del producto no siempre se encuentran bien documentados y en ocasiones no son siquiera reportados	Media	Serias	Documentar siempre los cambios que se realicen en la aplicación.
8	Planificación irreal	Organización		Se realizan planificaciones que están muy por encima de las posibilidades de los trabajadores o pueden ocurrir atrasos en el proyecto como resultado de no ajustarse a la planificación	Media	Serios	Ajustarse a la planificación del proyecto y no realizar planificaciones inalcanzables
9	Utilización de tecnologías inadecuadas	Herramientas		Las tecnologías son inadecuadas para la realización de las tareas, sea por desactualización de las mismas o por falta de recursos.	Media	Tolerable	Mantener actualizadas las tecnologías y cuidar las tecnologías existentes aplicándole mantenimientos regulares.
10	Falta de comunicación entre partes internas del proyecto.	Personal		El personal vinculado a la producción del producto tiene poca o nula comunicación.	Alta	Serias	El personal debe realizar sistemáticamente reuniones o talleres donde se analicen los resultados hasta el momento de su trabajo e

intercambiar
opiniones u
experiencias.

Tabla 13. Listado de Riesgos.

Anexo 4. Diagramas de clases del diseño con estereotipos web.

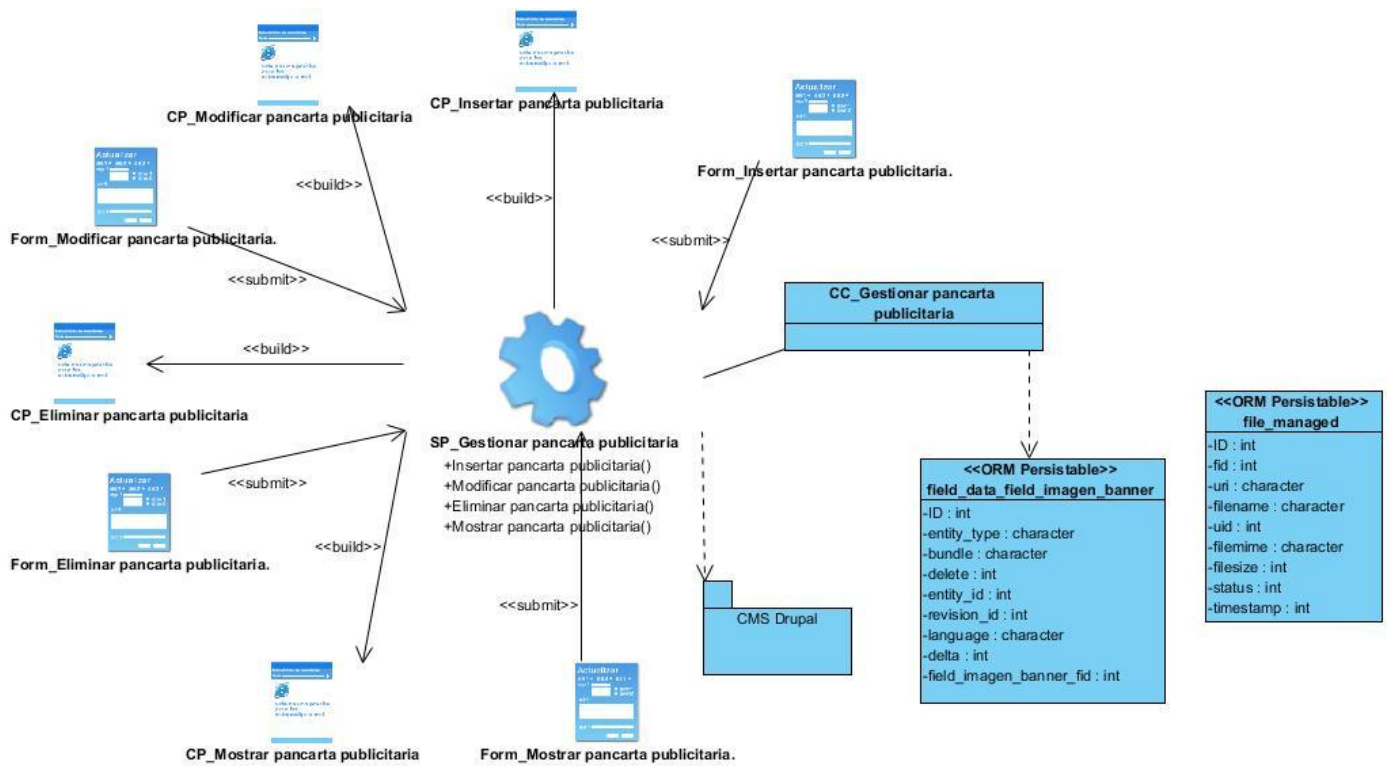


Figura 17. Diagrama de clases del diseño con estereotipos web del Gestionar pancartas publicitarias.

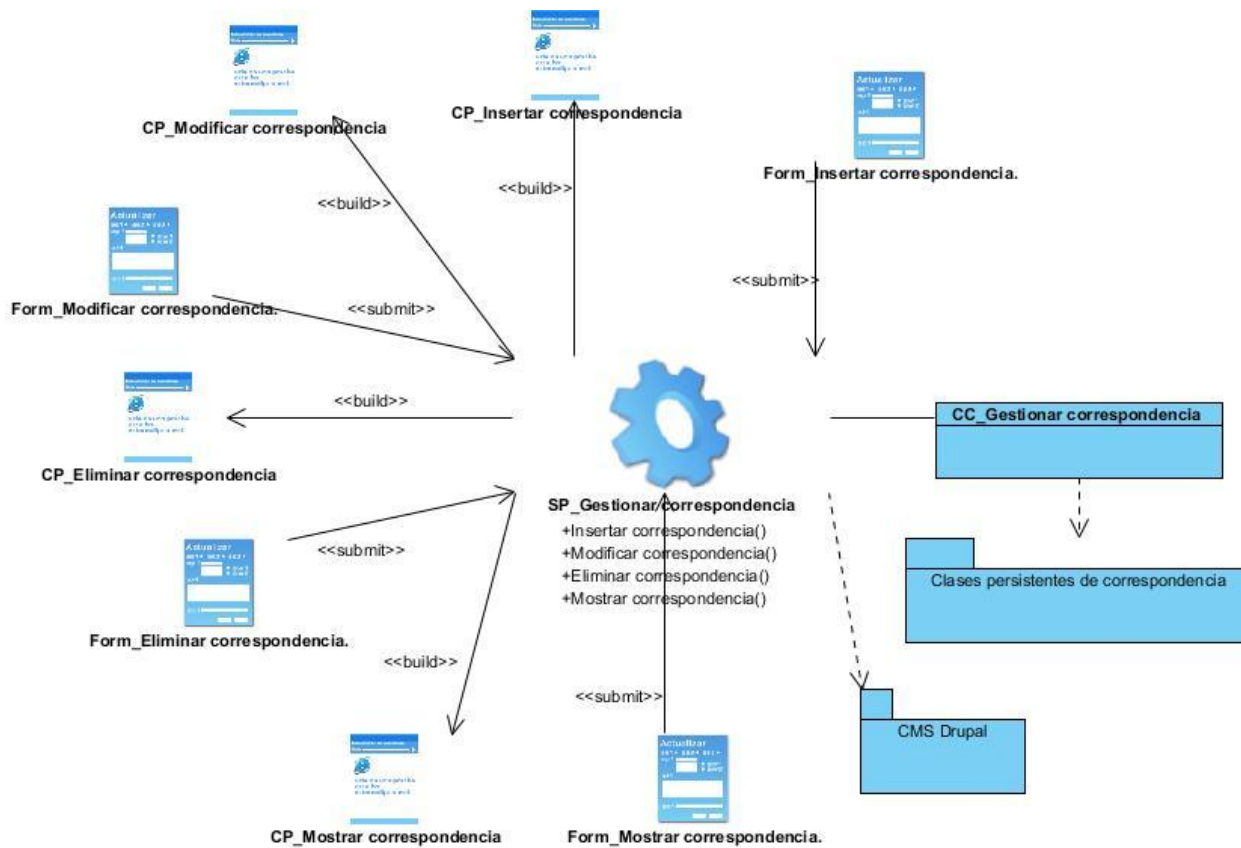


Figura 18. Diagrama de clases del diseño con estereotipos web del Gestionar correspondencia.

El paquete Clases persistentes de correspondencia se encuentra en el [Anexo 5](#).

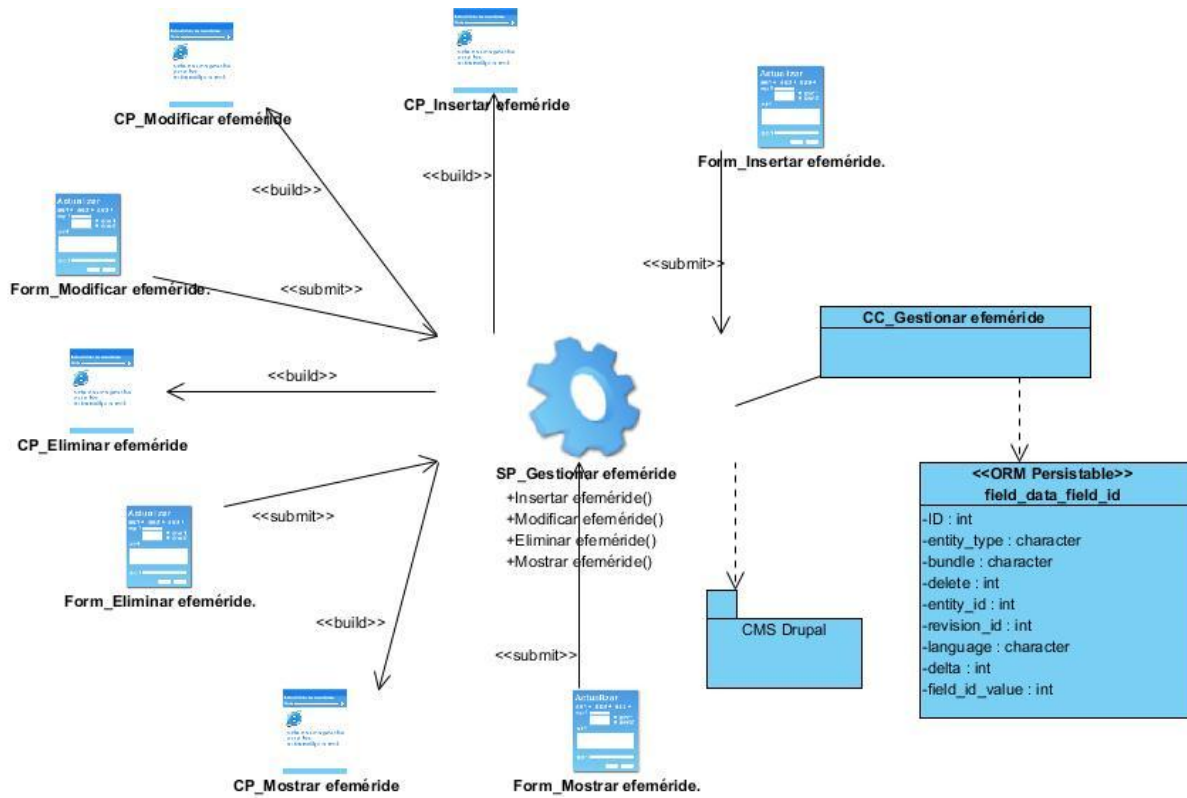


Figura 19. Diagrama de clases del diseño con estereotipos web del Gestionar efeméride.

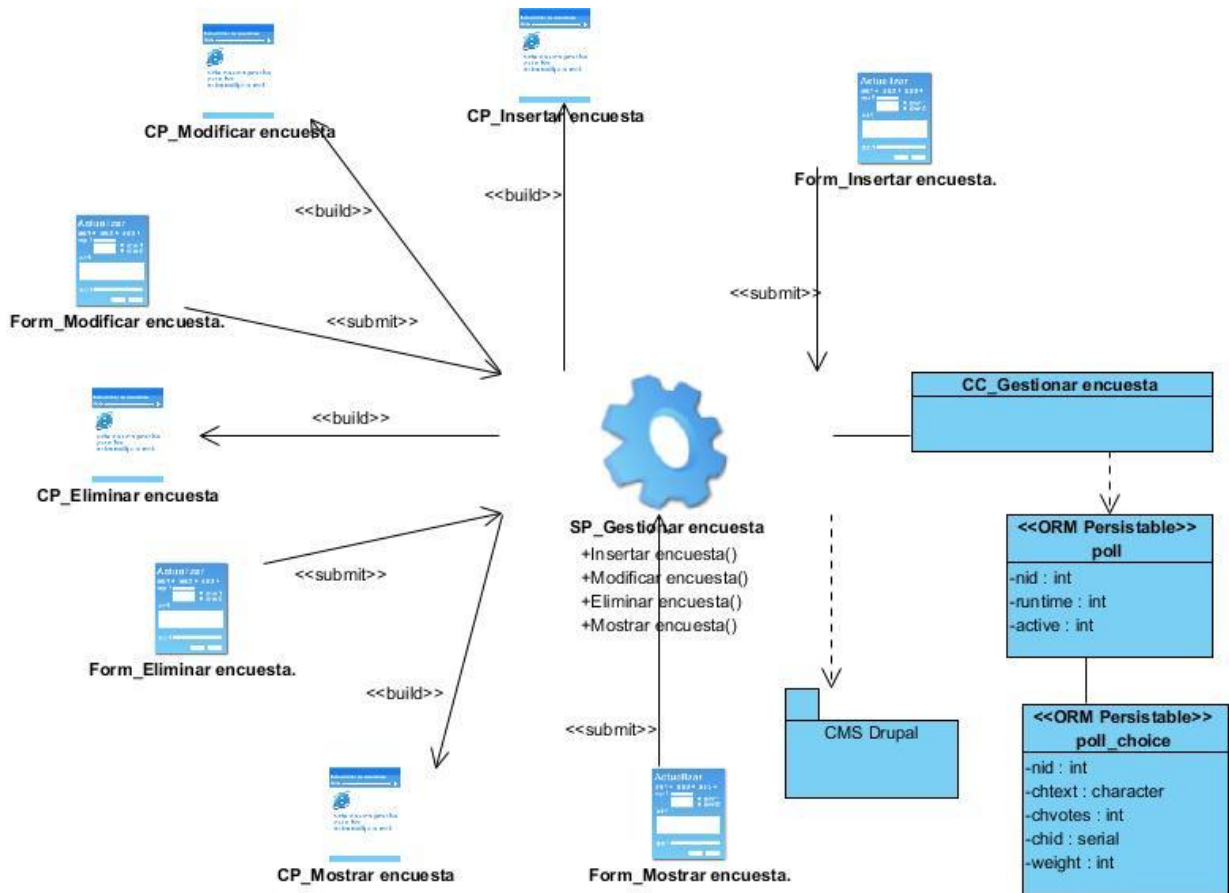


Figura 20. Diagrama de clases del diseño con estereotipos web del Gestionar encuesta.

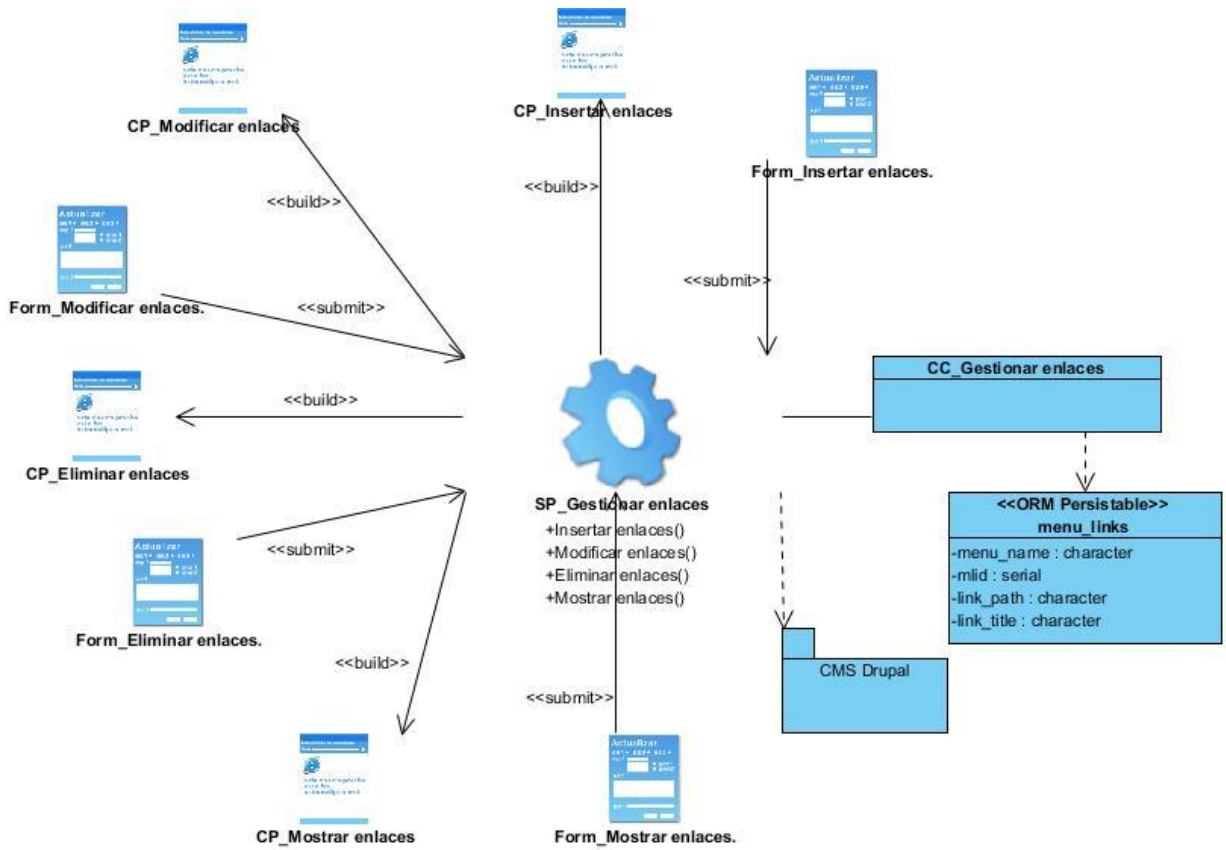


Figura 21. Diagrama de clases del diseño con estereotipos web del Gestionar enlaces.

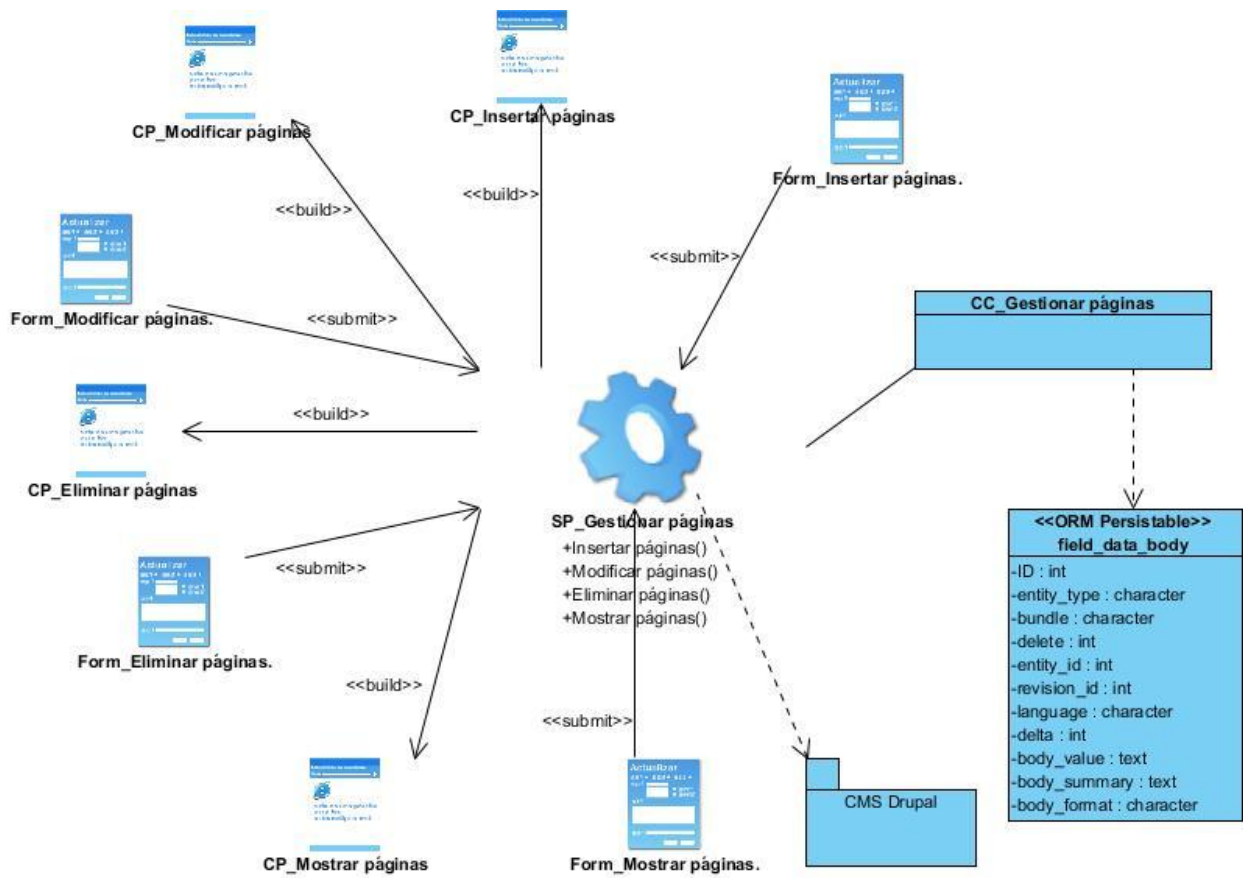


Figura 22. Diagrama de clases del diseño con estereotipos web del Gestionar página.

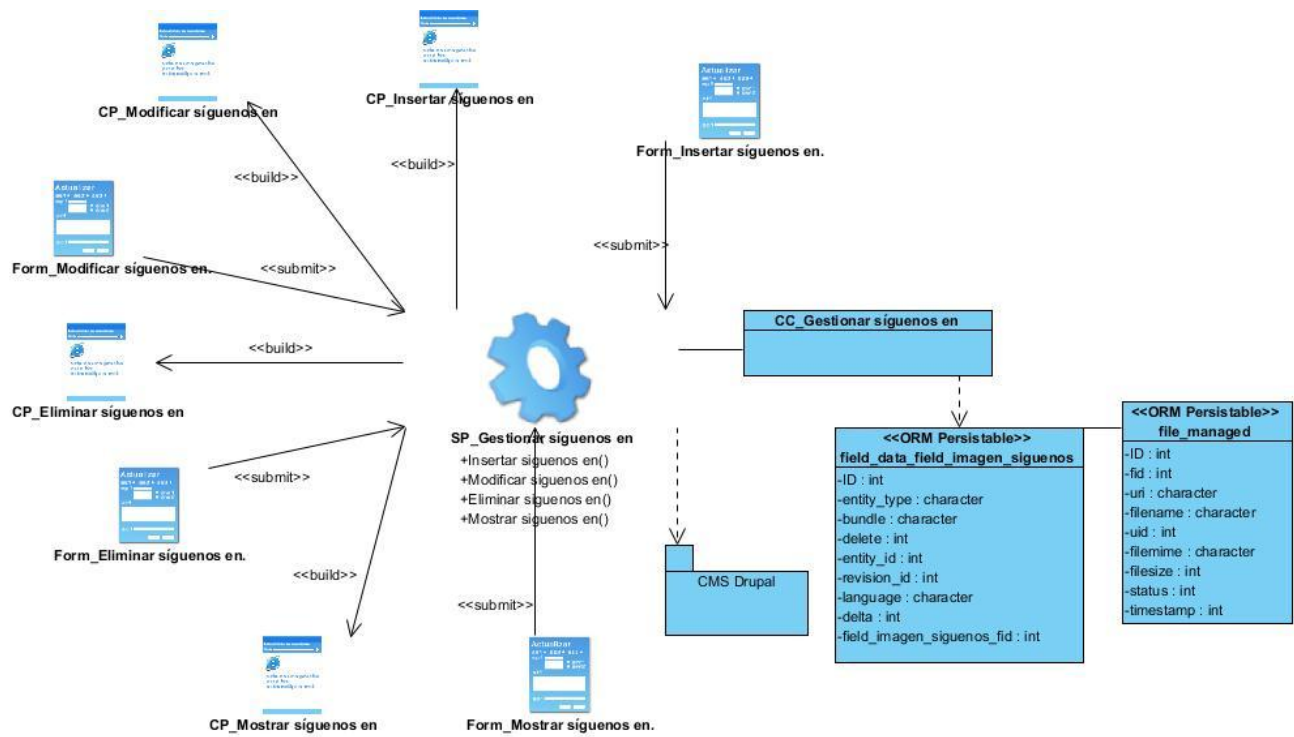


Figura 23. Diagrama de clases del diseño con estereotipos web del Gestionar siguenos en.

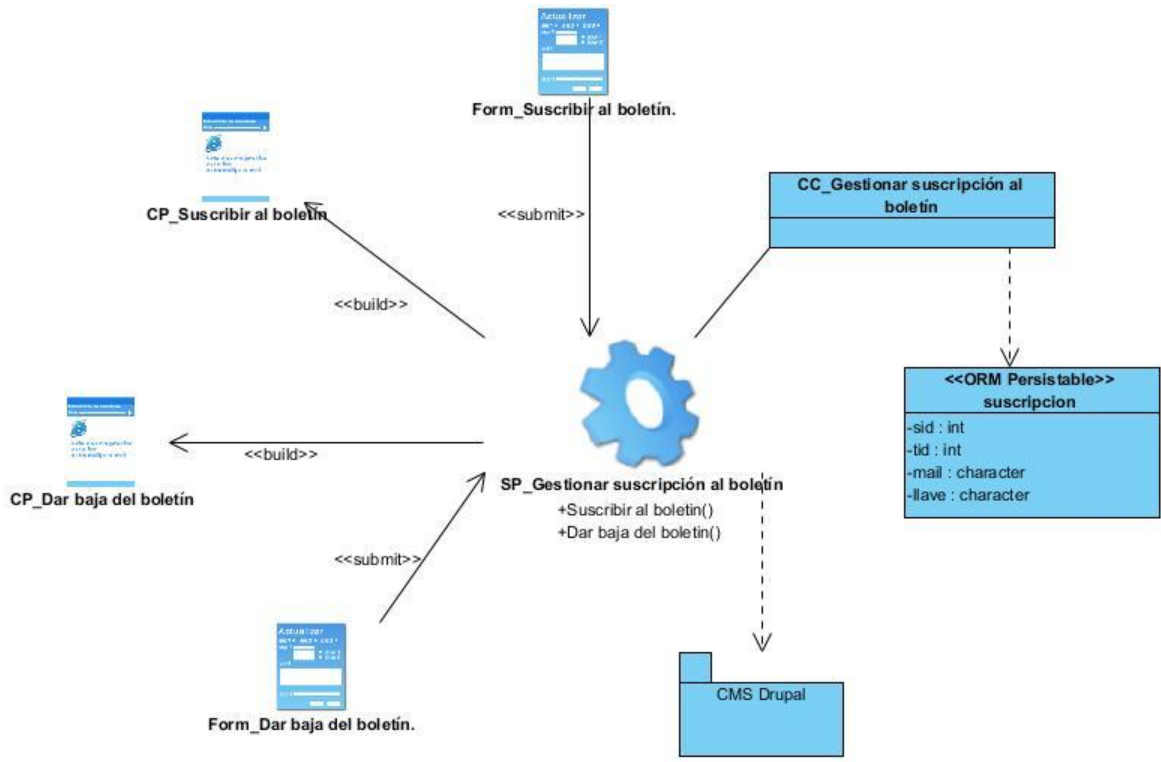


Figura 24. Diagrama de clases del diseño con estereotipos web del Gestionar suscripción al boletín.

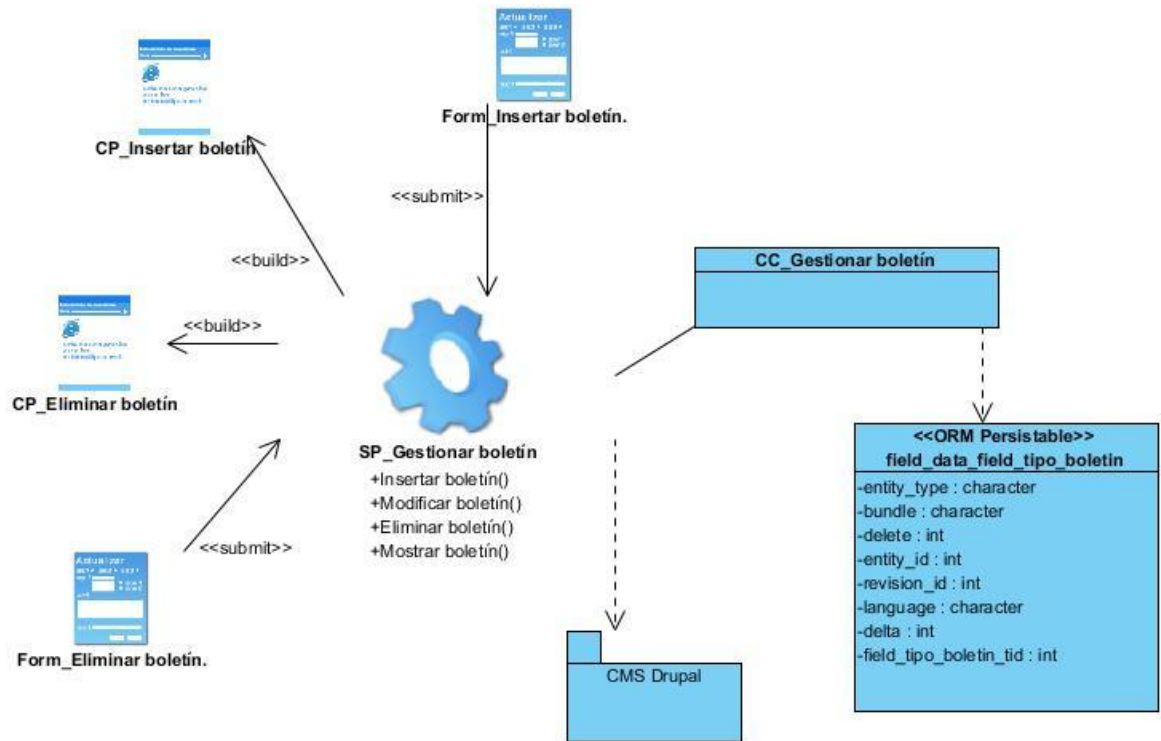


Figura 25. Diagrama de clases del diseño con estereotipos web del Gestionar boletín.

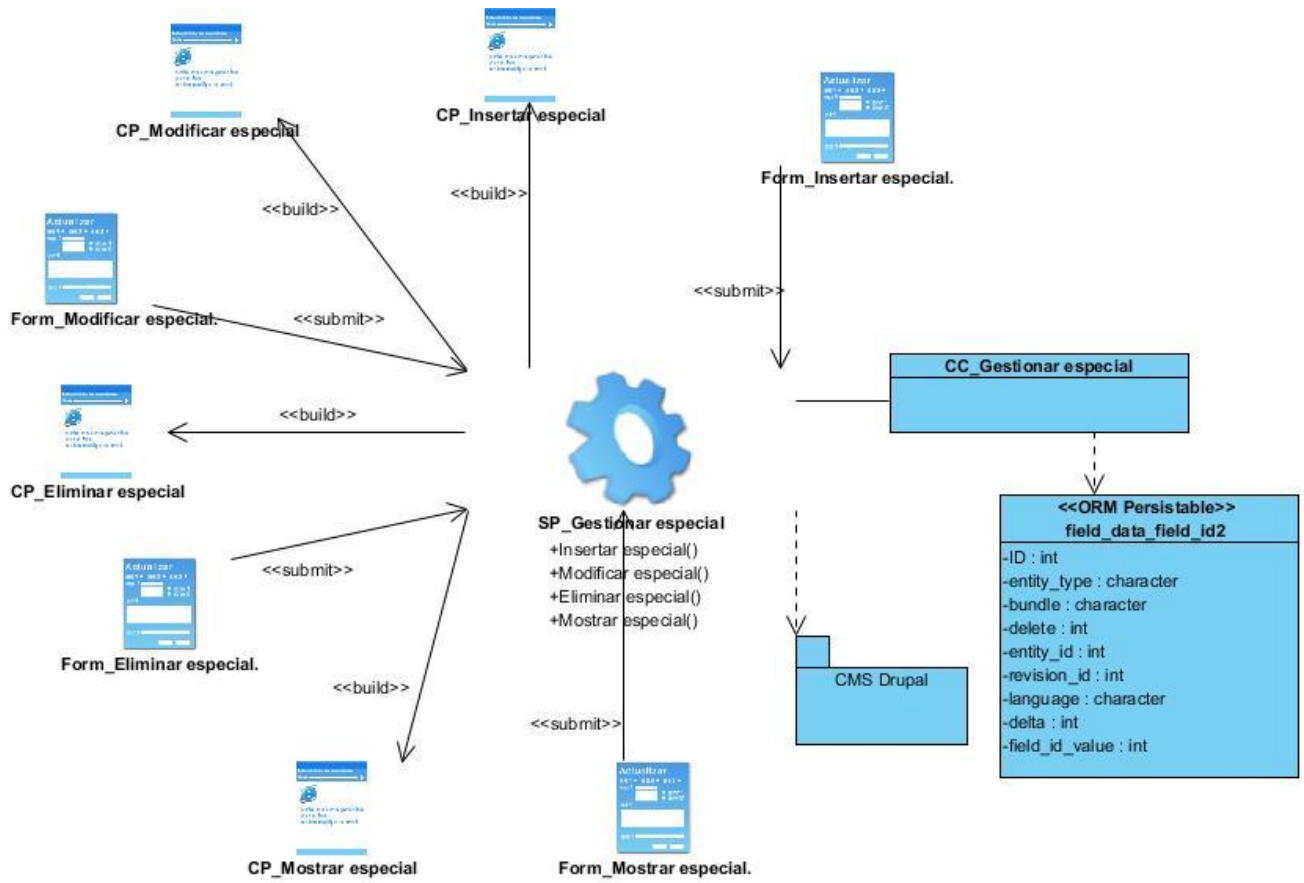


Figura 26. Diagrama de clases del diseño con estereotipos web del Gestionar especial.

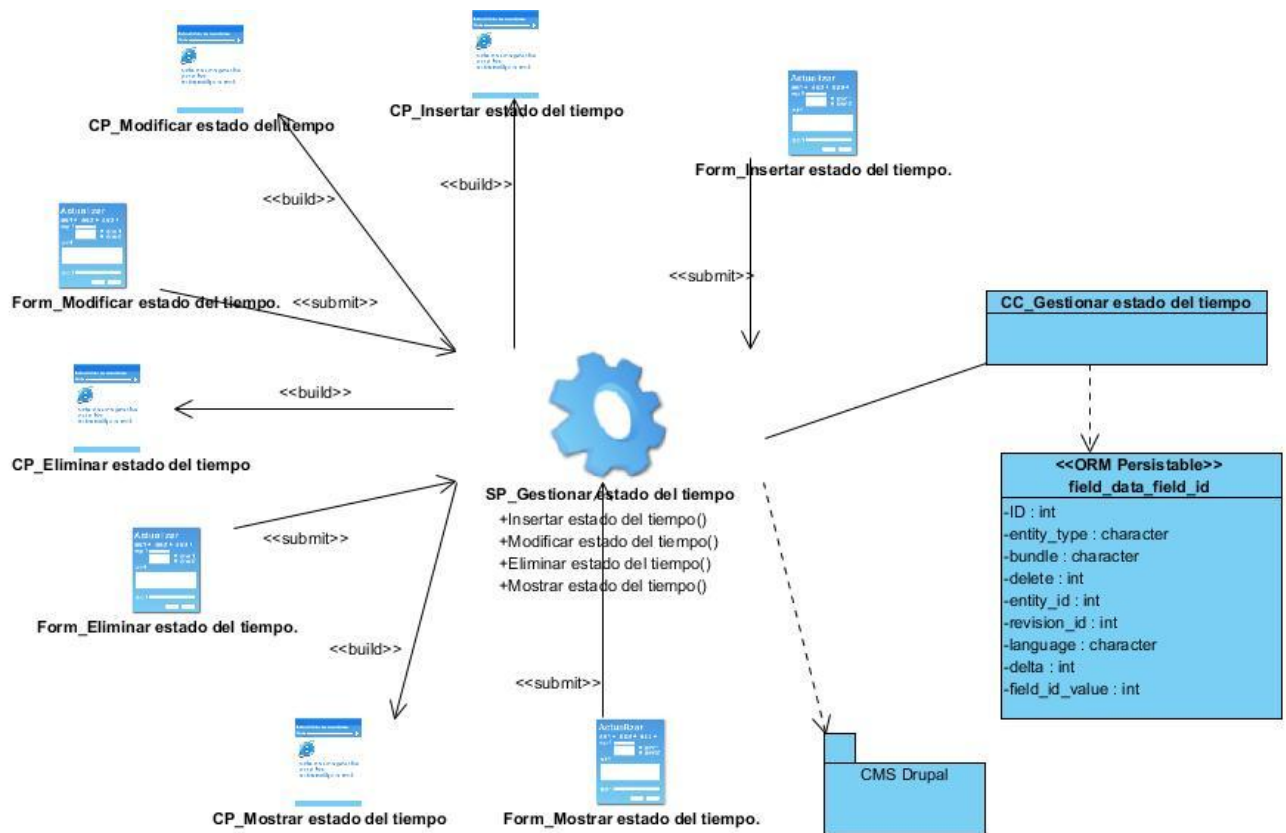


Figura 27. Diagrama de clases del diseño con estereotipos web del Gestionar estado del tiempo.

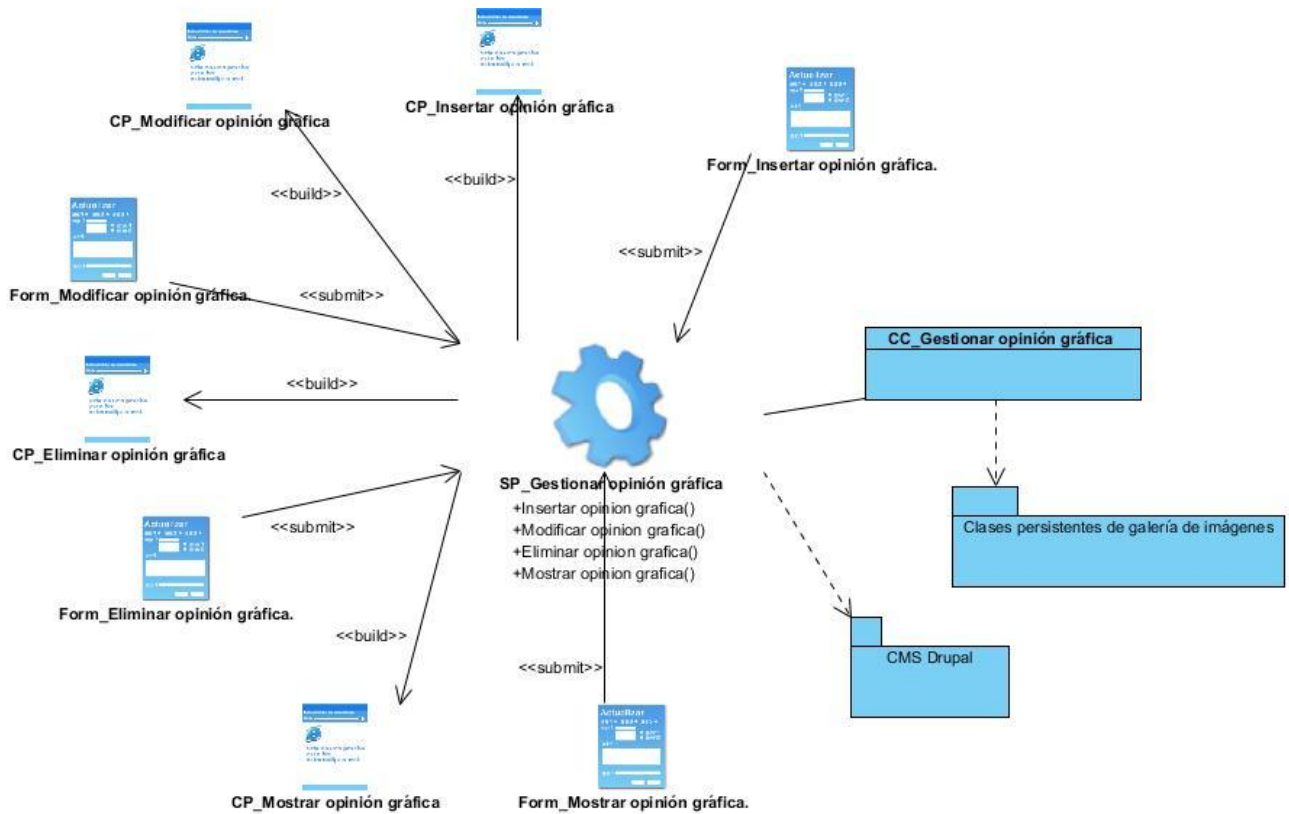


Figura 28. Diagrama de clases del diseño con estereotipos web del Gestionar opinión gráfica.

Anexo 5. Paquetes que contienen las clases persistentes.

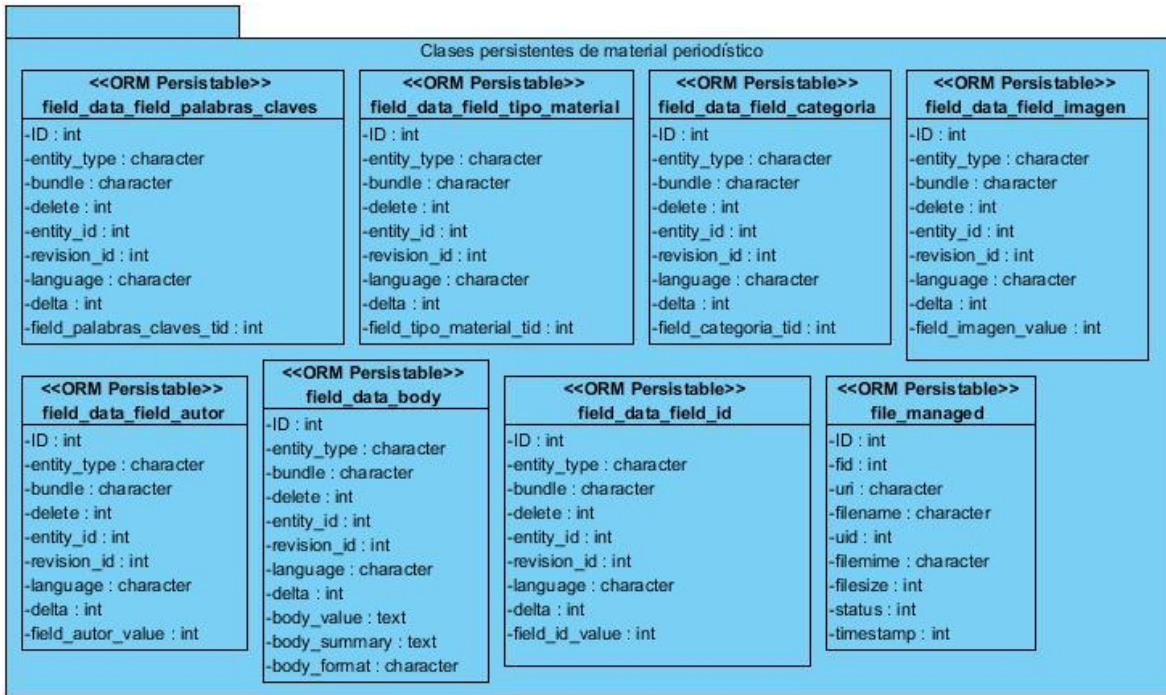


Figura 29. Diagrama de clases del diseño con estereotipos web.

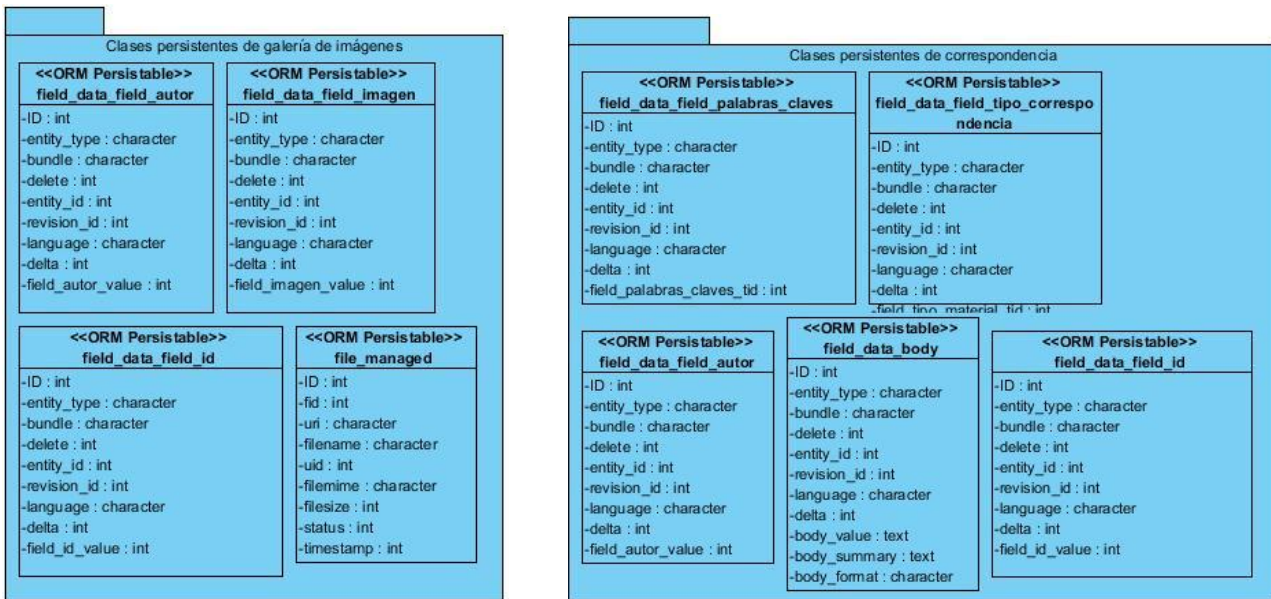


Figura 30. Diagrama de clases del diseño con estereotipos web.

Anexo 6. Descripción de las tablas.

Nombre: node		
Descripción: la tabla almacena los datos generales de un contenido		
Atributo	Tipo	Descripción
nid	int	Identificador primario para un nodo.
vid	int	Indica la última versión del nodo.
type	varchar	Tipo de nodo.
language	varchar	Lenguaje del nodo.
title	varchar	Título del nodo.
uid	int	Identificador de la tabla <i>users</i> que indica el usuario que creó el nodo.
status	int	Estado de publicación del nodo. Cero para los nodos “No publicados” y uno para los “Publicados”.
created	int	Fecha de creación del nodo.
changed	int	Fecha de la última modificación hecha al nodo.
comment	int	Especifica las opciones de comentario. Cero para “Deshabilitado”, uno para “Solo Lectura” y dos para “Lectura/Escritura”.
promote	int	Indica el grado de promoción del nodo, para que aparezca o no en la página principal.
sticky	int	Indica la prioridad del nodo sobre otros nodos promocionados.
tnid	int	La identificación del grupo de traducción de este nodo.
translate	int	Indica si esta página de traducción necesita ser actualizada.

Tabla 14. Descripción de atributos.

Nombre: field_data_field_id
Descripción: la tabla almacena los datos generales del campo id

Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de cada nodo.
field_id_value	int	Valor del id.

Tabla 15. Descripción de atributos.

Nombre: field_data_field_autor		
Descripción: la tabla almacena los datos generales del campo autor		
Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de cada nodo.
field_autor_value	int	Valor del autor.

Tabla 16. Descripción de atributos.

Nombre: field_data_field_palabras_claves		
Descripción: la tabla almacena los datos generales del campo palabras clave		
Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de cada nodo.
field_palabras_claves_value	int	Valor de las palabras clave.

Tabla 17. Descripción de atributos.

Nombre: field_data_field_fecha_edicion		
Descripción: la tabla almacena los datos generales del campo fecha edición		
Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de cada nodo.
field_fecha_edicion_value	int	Valor de la fecha de edición.

Tabla 18. Descripción de atributos.

Nombre: field_data_field_tipo_correspondencia		
Descripción: la tabla almacena los datos generales del campo tipo correspondencia		
Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de cada nodo.
field_tipo_correspondencia_value	int	Valor del tipo de correspondencia.

Tabla 19. Descripción de atributos.

Nombre: field_data_field_categoria		
Descripción: la tabla almacena los datos generales del campo categoría		
Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de

		cada nodo.
field_categoria_value	int	Valor de la categoría.

Tabla 20. Descripción de atributos.

Nombre: field_data_field_palabras_claves		
Descripción: la tabla almacena los datos generales del campo palabras clave		
Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de cada nodo.
field_palabras_claves_value	int	Valor de las palabras clave.

Tabla 21. Descripción de atributos.

Nombre: field_data_field_imagen		
Descripción: la tabla almacena los datos generales del campo imagen		
Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.

delta	int	Posición en la lista de id de cada nodo.
field_imagen_fid	int	Número del fid en la tabla file_managed que identifica a la imagen.
field_imagen_alt	varchar	Valor del alt de la imagen.
field_imagen_title	varchar	Valor del title de la imagen.
field_imagen_width	int	Valor del ancho de la imagen.
field_imagen_height	int	Valor del alto de la imagen.

Tabla 22. Descripción de atributos.

Nombre: file_managed		
Descripción: la tabla almacena los datos generales de un archivo		
Atributo	Tipo	Descripción
fid	varchar	Id del archivo.
uid	varchar	El identificador del usuario que está asociado al archivo.
filename	int	Nombre del archivo sin componentes de la ruta.
uri	Int	La ruta para acceder al archivo.
filemime	int	Tipo de archivo.
filesize	varchar	Tamaño del archivo.
status	int	Estado del archivo.
timestamp	int	Tiempo en que el archivo de añadió.

Tabla 23. Descripción de atributos.

Nombre: field_data_body		
Descripción: la tabla almacena los datos generales del campo cuerpo de texto		
Atributo	Tipo	Descripción
entity_type	varchar	Tipo de entidad (node, enlace, entre otros).
bundle	varchar	Tipo de contenido al cual pertenece.
deleted	int	Si está eliminado

		temporalmente.
entity_id	Int	Id del nodo al que pertenece.
revision_id	int	Id de la revisión del nodo al que pertenece.
language	varchar	Idioma del nodo al que pertenece.
delta	int	Posición en la lista de id de cada nodo.
body_value	text	Valor del cuerpo de texto.
body_summary	text	
body_format	varchar	Tipo de filtrado en el cuerpo de texto (HTML, PHP, entre otros).

Tabla 24. Descripción de atributos.

Anexo 7. Diagramas de componente.

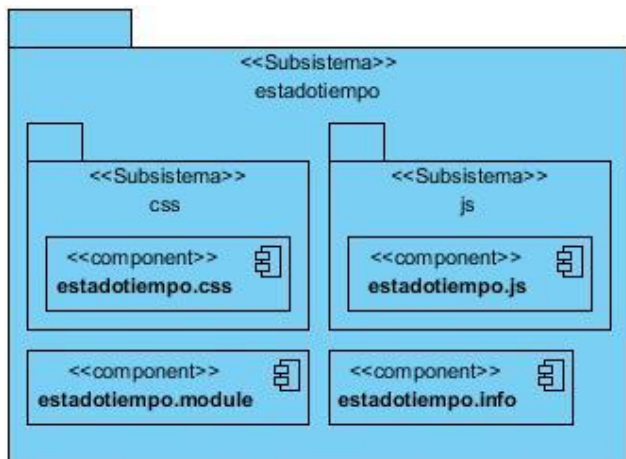


Figura 31. Diagrama de componentes del subsistema estadotiempo.

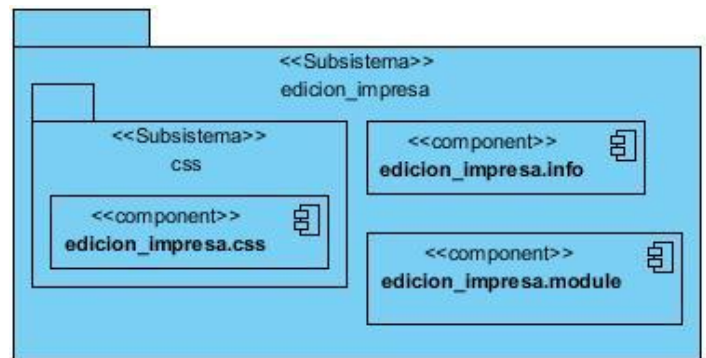


Figura 32. Diagrama de componentes del subsistema edicion_impresa.

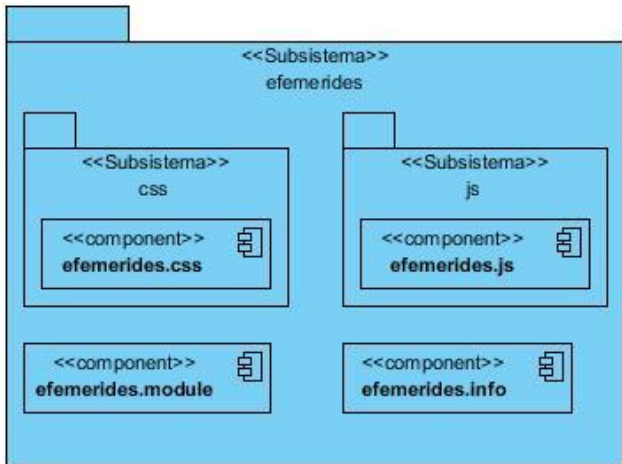


Figura 33. Diagrama de componentes del subsistema efemerides.

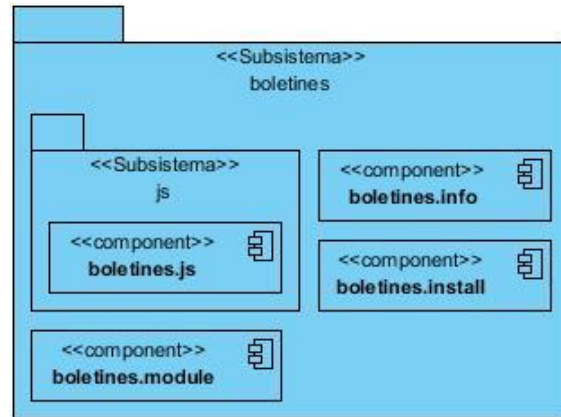


Figura 34. Diagrama de componentes del subsistema boletines.

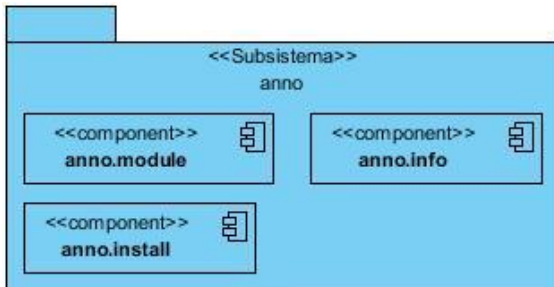


Figura 35. Diagrama de componentes del subsistema anno.

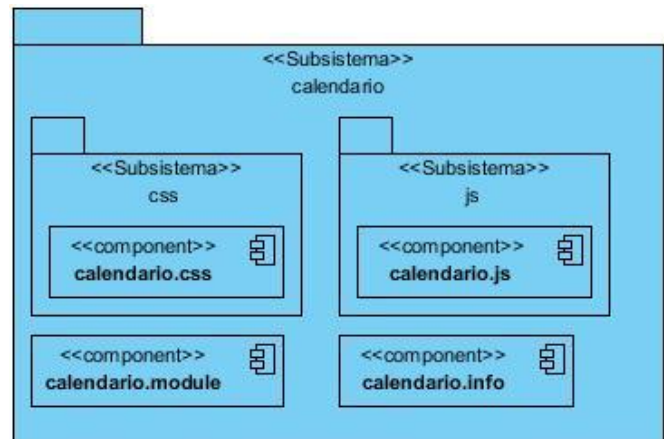


Figura 36. Diagrama de componentes del subsistema calendario.

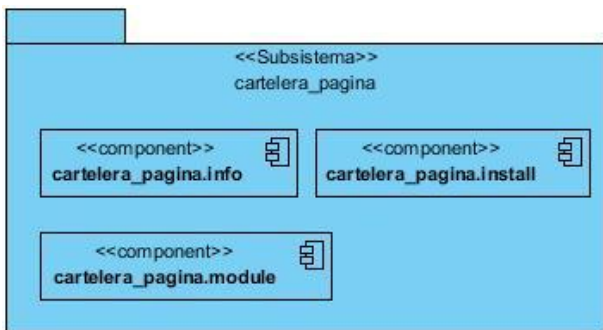


Figura 37. Diagrama de componentes del subsistema cartelera_pagina.

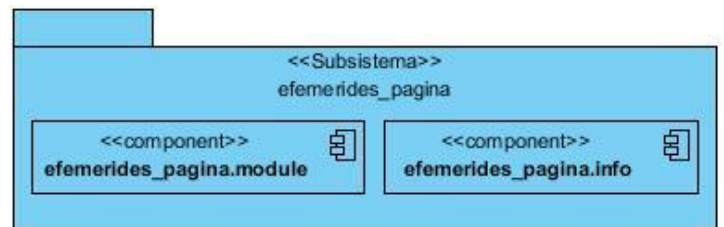


Figura 38. Diagrama de componentes del subsistema efemerides_pagina.

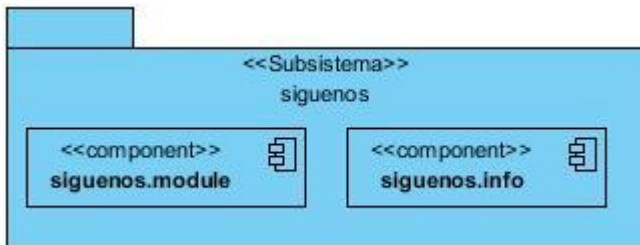


Figura 39. Diagrama de componentes del subsistema siguenos.

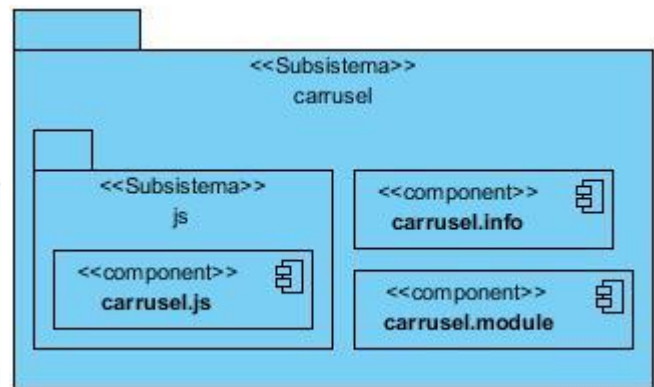


Figura 40. Diagrama de componentes del subsistema carrusel.

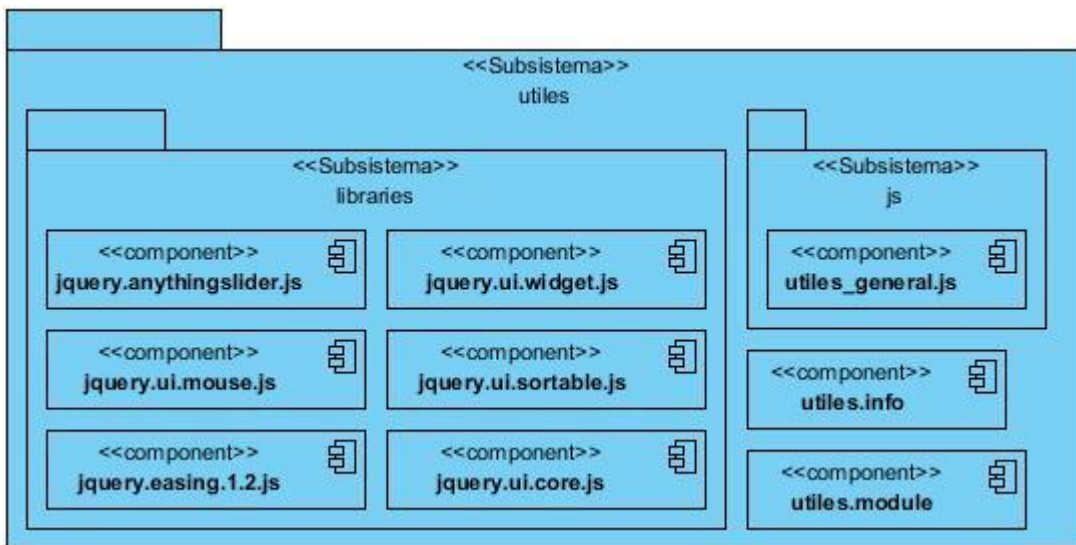


Figura 41. Diagrama de componentes del subsistema utiles.

Anexo 8. Casos de prueba de funcionalidad.

Se especifica cada uno por separado en los documentos contenidos en la carpeta “Casos de prueba de funcionalidad”, adjuntada al documento del trabajo de diploma.

Anexo 9. Casos de prueba de integración.

Se especifica cada uno por separado en los documentos contenidos en la carpeta “Casos de prueba de integración”, adjuntada al documento del trabajo de diploma.