

Universidad de las Ciencias Informáticas

Facultad 1



“Módulo de empaquetado web de contenidos para Drupal 7”

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

Autor:

Abel Salas López

Tutores:

Inst. Yuneldis Reyes Velázquez

Ing. Daniel Raúl Rodríguez Quintana

La Habana, Junio 2016



“Mi confianza en el triunfo final de lo que creo, es completa”

Ernesto Che Guevara

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al centro de Ideo-informática de la Universidad de las Ciencias Informáticas; así como a dicha institución 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 _____.

Abel Salas López

Firma del Autor

Ing. Yuneldis Reyes Velázquez

Firma del Tutor

Ing. Daniel Raúl Rodríguez Quintana

Firma del Tutor

Agradecimientos

Ha sido un largo camino el cual llega a su fin, pero no hubiera logrado llegar sin todos aquellos que me apoyaron en estos 5 años.

Primero gracias a mi familia por creer siempre en mí, incluso cuando ni yo creía que podía lograrlo. Gracias a mis profesores que me han tenido paciencia pues debo reconocer que soy un alumno inquieto. Gracias a mis amigos que han sido un soporte en el cual apoyarme cuando siento que el mundo se remueve bajo mis pies. Gracias a aquellos que ya no están conmigo, pero me dejaron un ejemplo a seguir y gracias a la UCI por brindarme estos 5 años llenos de experiencias únicas que nunca olvidaré.

Dedicatoria

En especial a mis padres que siempre soñaron con un hijo ingeniero y este es mi regalo para ustedes después de tanto que me han dado a mí en el transcurso de estos 24 años que llevamos juntos. También para Gerardo Fernández Menéndez que a pesar de no ser de mi sangre era otro padre para mí y aunque no esté a mi lado ha estado presente en cada paso de este camino recorrido.

Resumen

En el mundo existen países como Cuba que cuentan con poco acceso a la web por lo que se necesitan vías alternativas para la obtención y distribución de la información. El presente trabajo de diploma contiene una posible solución al problema señalado anteriormente, con el desarrollo de un módulo que permita obtener la información presente en sitios desarrollados sobre Drupal 7 y empaquetarla en forma de base de dato SQLite. Posteriormente se pretende distribuir la información empaquetada haciendo uso de las tecnologías móviles con una aplicación desarrollada para dispositivos con Android 4.1 o superior, que permita consumir la base de datos creada.

Con el fin de obtener los mejores resultados se trabaja bajo la guía de la metodología AUP-UCI, definiéndose como base tecnológica PHP y Java como lenguajes de programación y como entornos de desarrollo integrados NetBeans y AndroidEstudio.

Palabras clave: Android; *Drupal*, *módulo*; tecnologías móviles; *web*.

Índice de Contenido

Introducción	1
Capítulo 1: Fundamentación teórica del Módulo de empaquetado web de contenido para Drupal 7	7
1.1 Estudio de homólogos	7
1.2 Análisis de las tecnologías, herramientas y lenguajes a utilizar	9
1.3 Conclusiones parciales	20
Capítulo 2. Características del Módulo de empaquetado web de contenido para Drupal 7	21
2.1 Introducción	21
2.2 Modelo de dominio	21
2.3 Levantamiento de requisitos	23
2.5 Arquitectura	28
2.6 Diseño	31
2.7 Modelo de datos	33
2.8 Modelo de despliegue	35
2.8 Conclusiones parciales	35
Capítulo 3: Implementación y Prueba del Módulo de empaquetado web de contenido para Drupal 7....	37
3.1 Diagrama de componentes	37
3.2 Estándares de codificación	40
3.3 Pruebas	44
3.4 Conclusiones parciales	51
Conclusiones Generales	52
Recomendaciones	53
Referencias	54
Bibliografía Consultada	59

Glosario de términos.....	60
Anexos.....	63

Índice de Tablas

Tabla 1. Requisitos Funcionales.	24
Tabla 2. Descripción de HU: Seleccionar Campos por Tipo de Contenido.	27
Tabla 3. Descripción de componentes	39
Tabla 4. Tabla Estándar de codificación Drupal	40
Tabla 5. Estándar de codificación Java.	43
Tabla 6. Caso de pruebas: Escenario Seleccionar Tipo de Contenido.	44
Tabla 7. Caso de pruebas: Escenario Seleccionar Elementos del Menú.....	45
Tabla 8. Caso de Prueba: Escenario Enlazar Tipo de Contenido con Menú.	46
Tabla 8. Prueba de carga y estrés.	51
Tabla 9. Entrevista aplicada a directivos del centro CIDI.....	63

Índice de Figuras

Figura 1. Diagrama de Clases del Modelo de Dominio.	22
Figura 2. Arquitectura de Drupal (SeeD, 2015).	29
Figura 3. Diagrama de Clases de Diseño de Drupal.	32
Figura 4. Diagrama de clases de diseño. Seleccionar Campo por Tipo de Contenido.	33
Figura 5. Modelo relacional.....	34
Figura 6. Diagrama de Despliegue.....	35
Figura 7. Diagrama de Componentes.	38
Figura 8. Resultados de las pruebas funcionales.	48
Figura 9. Resultados de las pruebas de seguridad.	49

Figura 10. Diagrama de clases de diseño. Seleccionar Tipo de Contenido y Seleccionar Menús.	63
Figura 11. Diagrama de clases de diseño. Seleccionar Campos por Tipo de Contenido.....	64
Figura 12. Diagrama de clases de diseño. Enlazar Tipo de Contenido con Menú.....	64
Figura 13. Diagrama de clases de diseño. Aplicación.....	65
Figura 14. Seleccionar Tipos de Contenidos y Menús.	66
Figura 15. Campo del Tipo de Contenido a exportar.....	66
Figura 16. Enlazar Tipos de Contenidos y Menús.....	67

Introducción

Con el surgimiento de la *World Wide Web* (WWW) o la web como es conocida comúnmente se desata un flujo de información global a una escala sin precedentes en la historia humana. La web, sorteó las barreras creadas por la geografía permitiendo que personas separadas en el tiempo y el espacio, puedan intercambiar o incluso desarrollar mutuamente nuevos datos, demostrar sus actitudes y deseos cotidianos, experiencias emocionales, cultura, idiomas, negocio, todo tipo de información puede ser compartida y diseminada digitalmente con el menor esfuerzo (Lapiente, 2013). Por su carácter virtual, los datos no ocupan un gran espacio físico como en una biblioteca, permitiendo que la información en la web pueda ser buscada más fácil y eficientemente que en cualquier medio físico, y mucho más rápido de lo que una persona podría recabar por sí misma a través de un viaje, correo, teléfono, telégrafo o cualquier otro medio de comunicación.

Según (Oliva, 2009) tener la información justa en el momento oportuno significa poder y solamente las personas o grupos de individuos que se puedan permitir financiar este intercambio de información disfrutaran de este privilegio. Manuel Castells introduce una idea interesante sobre la relación entre información y poder, diciendo que socialmente, la información suele estar ligada al poder. De esta manera aquellos usuarios que no acceden o que no logran acceder a esta información están en clara desventaja frente a aquellos que la tienen a su disposición. Evidenciando la necesidad y ventaja que representa tener acceso a una fuente de información que crezca y se actualice rápidamente como es la web.

Para analizar la repercusión de la web en la sociedad se selecciona Cuba como país con poco acceso a la información presente en ella. No fue hasta 1991 que la isla participa por primera vez en un evento de Internet, que contó con los principales actores de América Latina y el Caribe, incluyendo a Estados Unidos y Canadá. En este mismo año se registra por primera vez un dominio, en este caso el hosting .cu. El 22 de agosto de 1996, Cuba queda conectada a la Internet con el proveedor internacional SPRINT CORP, compañía norteamericana, por un canal satelital a 64k. Para esto fue necesario hacer el traslado del dominio .cu y ponerlo a nombre de CENIAI, entidad encargada de proveer los servicios en la isla (Schlachter, 2008).

Cuba es un país que ha sufrido un bloqueo económico por más de medio siglo provocando dificultades económicas y tecnológicas. Se puede apreciar claramente esta cruel política al observar los cables de fibra óptica que rodean la isla y no tocan territorio nacional. Las dificultades también se ven en los problemas

presentados por la empresa cubana ETECSA, esta se ha visto imposibilitada de adquirir equipos y piezas de repuesto cuya patente es de compañías norteamericanas y que resultan imprescindibles para el desarrollo y mantenimiento de las telecomunicaciones cubanas. Estos componentes son adquiridos en el mercado internacional bajo fuertes medidas de control y supervisión, además de un incremento considerable de su valor.

A principios del 2015 Cuba según datos de la Unión Internacional de Telecomunicaciones, organismo de la ONU con sede en Ginebra, sólo el 3,4 por ciento de los hogares cubanos están conectados a Internet y más del 80 por ciento de los cubanos no tienen acceso regular (UIT, 2015). Evidenciándose el poco acceso a Internet que presenta la isla, aunque se realizan ingentes esfuerzos para crear nuevas estrategias que permitan mayor acceso de la población a la web.

Actualmente se están facilitando puntos de acceso a Internet a la población mediante la puesta en servicios de zonas Wi-Fi. Esta alternativa ha logrado gran aceptación por parte de la población, pero el acceso es limitado por la disponibilidad del servicio teniendo en cuenta la cantidad de usuarios presentes en la isla. También presenta como desventaja que los usuarios tienen que trasladarse hacia estos puntos para obtener sus servicios no pudiendo acceder desde casas o centros laborales. Todo lo planteado anteriormente limita a los clientes de acceder a la información que precisan regularmente (ETECSA, 2015).

En el país existen pequeñas y medianas empresas que poseen una débil infraestructura tecnológica, donde sus trabajadores no cuentan con acceso a la red nacional e internacional. Conociendo de la ventaja que representaría para ella tener acceso a la información presente en la red referente al negocio.

Por otra parte, se encuentran regiones de poca accesibilidad para el hombre y la tecnología, donde todavía queda mucho camino por recorrer antes de lograr la llegada del Internet a los mismos. Lo que supone un bloqueo de la información, la cual es un valioso recurso para el desarrollo de estas regiones. Esto genera un lento y trabajoso desarrollo de estas zonas.

Sin embargo, dentro de Cuba se observa una penetración sin precedente de la tecnología móvil, principalmente con Sistema operativo Android. Según informa la empresa de telecomunicaciones cubana ETECSA en abril del 2015 se sobrepasó la cifra de tres millones de usuarios con líneas móviles. La misma empresa reconoce que, aunque no existen las estadísticas exactas del uso de dispositivos móviles con sistema Android esta es muy elevada sobre todo en jóvenes de 11 a 26 años. Estos dispositivos permiten

la interacción de diversos usuarios y la explotación de la distribución de la información *offline* con aplicaciones como el Zapia. Observando estas características que existen dentro de la isla aumentaría la divulgación de la información si se pudiera llevar la información a estos dispositivos y permitirle la navegación y distribución *offline*.

Vistas las circunstancias vigentes en la actualidad una vía para la solución de los problemas establecidos sería la distribución de la información *offline*. Permitiendo el acceso a la información presente en el espacio virtual por usuarios sin acceso a Internet.

En la Universidad de las Ciencias Informáticas (UCI) existen diferentes facultades, cada una de ellas desarrolla un perfil diferente y tienen asociados centros de desarrollo. El Centro de Ideoinformática (CIDI) perteneciente a la Facultad 1, tiene como una de sus misiones proveer soluciones, productos y servicios relacionados con las tecnologías de Internet, en función defender la ideología socialista en la red de redes. Este centro trabaja sobre el CMS Drupal para la creación de portales web.

Drupal, es un sistema de gestión de contenidos que se utiliza para crear sitios web dinámicos y con gran variedad de funcionalidades. Con este CMS es posible implementar una gran variedad de sitios web: un blog personal o profesional, un portal corporativo, una tienda virtual, una red social o comunidad virtual. Actualmente más de 1 millón de sitios web usan Drupal lo que representa el 6.7% de los sitios mundiales y el 12% de los 10 000 sitios web mejor posicionado (Drupal, 2013).

Según lo planteado anteriormente se llegó a la conclusión de que en Cuba existen grandes problemas en el acceso a Internet lo que resulta en que menos del 20 por ciento de la población de la isla accede al mismo regularmente. Existe un abundante aumento del uso de las tecnologías móviles con Sistema operativo Android. Se tiene a Drupal como uno de los CMS más usados para la creación de sitios web, además, en Cuba es muy común el uso de esta tecnología de desarrollo para construir sitios web lo cual representa un gran cúmulo de información virtual. Además, existen zonas de bajo desarrollo las cuales le sería muy útil una forma de acceso a la información virtual, así como empresas e instituciones a las cuales la información actualizada de la red le sería de gran beneficio.

Teniendo en cuenta lo anteriormente expuesto se plantea el siguiente **problema de investigación**: ¿Cómo contribuir a la divulgación de la información publicada en sitios web sobre Drupal 7 para usuarios sin conexión a Internet?

Se define como **objeto de estudio** el proceso de divulgación de la información y se enmarca la investigación en el **campo de acción** definido por el conjunto de herramientas para la divulgación de la información para usuarios sin conexión a Internet.

El **objetivo general** que se plantea es desarrollar una solución informática compuesta por un módulo que permita el empaquetado del contenido publicado en sitios web sobre Drupal 7 y una aplicación para el sistema operativo Android que lea dichos paquetes y muestre el contenido al usuario.

Como **objetivos específicos** se definen:

1. Describir el estado del arte referente a herramientas para la divulgación de la información para usuarios sin conexión a Internet.
2. Identificar las funcionalidades necesarias y realizar el diseño de la solución informática a desarrollar.
3. Implementar las funcionalidades de la solución informática a desarrollar.
4. Validar la solución informática a desarrollar a través de pruebas de *software*.

El presente trabajo tiene como **idea a defender que** con el desarrollo de la solución informática constituida por un módulo de empaquetado web de contenidos publicados en sitios web sobre Drupal 7 y una aplicación de Android para mostrar dicho contenido, se contribuirá a la divulgación de la información para usuarios sin conexión a Internet.

Para dar cumplimiento a los objetivos trazados anteriormente se efectuarán las siguientes **tareas de investigación**:

1. Descripción del estado del arte sobre herramientas para la divulgación de la información para usuarios sin conexión a Internet.
2. Levantamiento de información para identificar los requisitos de *software*.
3. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
4. Diseño de la arquitectura, selección de patrones de diseño y arquitectura.
5. Implementación de la solución propuesta.
6. Diseño de pruebas de *software* para validar la solución implementada.

7. Validación de la solución implementada.

Los métodos de investigación utilizados en la presente investigación son:

Métodos teóricos:

El método Analítico-Sintético se utilizó en la búsqueda y análisis de herramientas para la divulgación de la información para usuarios sin conexión a Internet. También en el análisis de la estructura de Drupal y la organización de sus contenidos, menús y nodos para lograr una mayor comprensión del sistema, analizando cómo se relacionan sus componentes entre sí.

El método Histórico-Lógico permitió una mayor comprensión del estado y tendencias actuales del proceso de divulgación de la información para usuarios sin conexión a Internet. Se utilizó para analizar y estudiar los antecedentes y evolución de los elementos asociados al desarrollo de herramientas para la divulgación de la información.

La Modelación permitió una mejor comprensión de los elementos analizados y diseñados para posterior implementación mediante el uso de diagramas y modelos más simples.

Métodos empíricos:

La Observación fue usada como método para apreciar cómo se estructura y distribuye la información dentro de los sitios desarrollados en Drupal y también como se le hace llegar a los usuarios registrados en el mismo.

La Entrevista se empleó en encuentros con profesores, especialistas y el jefe de departamento de Servicios Informáticos para Internet del centro CIDI, la cual permitió entender mejor el proceso de creación de la información y su clasificación. Además, como se implementan los portales dentro del centro, que tecnologías usan y para definir requisitos que se identificaron en esas entrevistas. **Ver anexo N° 1**

Se definió que se podía obtener la información presente en los portales a través de los nodos almacenados en la base de datos, que esta información se almacenaba mediante estructuras denominadas “Tipos de contenidos” de los cuales existían una o muchas instancias. Estos tipos de contenidos están compuestos por texto, imágenes o archivos, estos elementos se ordenan de acuerdo a las necesidades del usuario. Se

determinó que Drupal es independiente del Sistema Gestor de Base de Datos y el Servidor web y utiliza como lenguaje de programación principal PHP. Se definieron como requisitos la necesidad de definir roles y control de usuarios para el trabajo con el módulo.

La presente investigación está estructurada en 3 capítulos de la siguiente manera:

Capítulo 1. Fundamentación teórica del Módulo de empaquetado web de contenido para Drupal 7:

En este capítulo se definen conceptos importantes para la comprensión de la investigación. Se realiza un estudio y valoración sobre algunas de las soluciones existentes del proceso de divulgación de la información y empaquetado web en Cuba y el mundo, y se explican las herramientas, metodologías y lenguajes que serán utilizados en la construcción de la solución.

Capítulo 2. Características del Módulo de empaquetado web de contenido para Drupal 7:

En este capítulo se presenta y analiza la solución que se propone, se brindan elementos de cómo está concebido el negocio y se realiza la selección de los requerimientos del sistema que se pretende implementar. Se modelan y describen los diagramas que representan las funcionalidades del sistema a construir, aplicando los patrones de arquitectura y diseño seleccionados.

Capítulo 3: Implementación y Prueba del Módulo de empaquetado web de contenido para Drupal 7:

En este capítulo se describen la implementación, y posterior validación realizada al producto obtenido como solución.

Como **posible resultado** se obtendrá una solución informática compuesta por un módulo que permita el empaquetado de la información contenida en los sitios web desarrollados con Drupal 7 en una base de datos y una apk capaz de consumir esta base de datos permitiendo acceder a la información anteriormente empaquetada desde dispositivos con Sistema operativo Android, contribuyendo así a la distribución de la información offline.

Capítulo 1: Fundamentación teórica del Módulo de empaquetado web de contenido para Drupal 7

En el presente capítulo se hace un análisis de los principales conceptos asociados a la investigación para una mejor comprensión como: “navegación *offline*”, “apk para teléfonos Android” entre otros. Se definen las herramientas a utilizar, así como sus características y motivos para su uso, se define y describe la metodología de desarrollo de *software* a utilizar, así como sus fases, flujos de trabajo y artefactos que se proveen. Se realiza un análisis de las soluciones existentes, justificando la negativa a usar alguna de estas y definiendo características reutilizables, así como las posibles soluciones para el mismo y se define cual se seleccionará debido a sus facilidades.

1.1 Estudio de homólogos

En la búsqueda de los antecedentes en cuanto a herramientas para la distribución o consumo de información *offline*, se analizan softwares como *HTML Export* y *Static Generator* como módulos del CMS Drupal. En aplicaciones de escritorio se selecciona para su análisis *HTTrack* como una buena opción a explorar para el empaquetado de portales. Además, en el ámbito nacional se hallan alternativas que hacen uso de la navegación *offline* explotando las crecientes tecnologías móviles existentes en el país.

Internacionales

HTML Export:

Exportación HTML permite seleccionar un sitio desarrollado sobre Drupal 7 y elegir caminos del mismo en base a criterios para exportar a HTML. Mediante el uso de los *hook* es capaz de añadir o definir o modificar los caminos, activos y datos del HTML utilizado en la exportación (Ollendyke, 2014).

1. Soporta diferentes rutas para exportar entre ellas por nodos.
2. Criterios de selección de *hook* mediante las API de Drupal o por vistas.
3. Como desventaja se encuentra que no existe una versión estable para Drupal 7 ya que sigue en desarrollo.

Static Generator:

Genera una copia completa del sitio web en forma HTML incluyendo todos los js, CSS, imágenes y otros activos. La cual puede ser transferida y ejecutar el sitio web de un servidor web simple sin PHP, MySQL o memcached (Knutson, 2015). A continuación, se observarán las principales características de este módulo:

1. Aumenta el rendimiento pues al ser un sitio estático y sin la presencia de PHP no necesita hacer consultas a ningún servidor y aumenta la respuesta del sistema.
2. Más seguro ya que al no poseer bases de datos no almacena ninguna información referente a los usuarios o cualquier interacción de estos con el sistema.
3. La mayor parte de los puntos de fallos habituales se encuentran en consultas de PHP o SQL, los cuales ya no forman parte del sistema, por lo que convierte en un sitio más confiable.
4. Debido al hecho de que no hay ninguna base de datos o PHP detrás de la página web, cualquier funcionalidad que los requiera no se encontraran activas en el sitio. Lo que puede ser reconocida como su principal desventaja.

HTTrack

Es un *software* libre y de fácil uso que permite hacer uso de la navegación *offline*. El mismo permite la descarga de un sitio web a un directorio local, construyendo recursivamente todos los directorios, descargando el código HTML, las imágenes, CSS, js desde el servidor a su computadora: Este *software* crea una estructura de enlaces relativos igual a la del sitio original. Basta con abrir una página del sitio web espejo en su navegador, y se puede navegar por el sitio de enlace a enlace, como si se estuviera viendo en línea. También puede actualizar descargado anteriormente, y reanudar descargas interrumpidas, es totalmente configurable, y tiene un sistema de ayuda integrado (Kauler, 2015).

Nacionales

En el ámbito nacional se pudo detectar que existe un gran flujo de la información virtual por el país sobre todo en los últimos años. Actualmente en el país se encuentran sistemas de distribución “*offline*” de contenidos televisivos extranjeros, programas, música y otros productos digitales que se denomina El Paquete. Dentro de la isla se pueden destacar aplicaciones de sistemas Android como “Zapia” o “Conoce Habana” las cuales no necesita de conexión a Internet para brindar sus servicios, esta última brinda mapas de la capital cubana e información de lugares de interés como restaurantes, cines, monumentos y muchos

más. Se aprecia el auge de la navegación y distribución de la información *offline* en la isla, sin embargo, no se pudo encontrar ninguna aplicación que permitiera el empaquetado web.

1.1.1 Resultados obtenidos del estudio

Como resultado del estudio realizado se concluye que no constituyen una solución factible ninguna de las aplicaciones de empaquetado web analizadas, no existe una aplicación que permita obtener la información presente en la web y distribuirla sin conexión explotando las tecnologías móviles existentes en el país. Además, las herramientas de empaquetado de contenido no permiten a los usuarios la selección y configuración de la información que desean obtener, obligando a los usuarios a obtener lo que podría ser clasificados datos basura o información no útil al usuario. No obstante, se identificaron características y componentes reutilizables en el desarrollo de la solución, como serían el uso de los *hook* y nodos para crear caminos mediante los cuales se exportarán los diferentes recursos de sitio y la exclusión del lado del servidor para aumentar el rendimiento, confiabilidad y seguridad. Como otra característica se obtiene la posibilidad de distribuir la información a través de aplicaciones para Android aprovechando el auge con que cuentan en estos momentos asegurando una mayor divulgación de la información descargada.

1.2 Análisis de las tecnologías, herramientas y lenguajes a utilizar

En este epígrafe se analizan las tecnologías, herramientas y lenguajes a utilizar para el desarrollo de la solución.

1.2.1 Lenguajes utilizados

HTML5: es un lenguaje de marcado empleado para estructurar y presentar contenido en la web. Como su nombre lo indica es la quinta revisión del estándar HTML y permite soportar multimedia, agrega elementos como video, audio y canvas, como la integración para gráficos vectoriales (SVG) y MathML para fórmulas matemáticas. Estas características permiten incluir y controlar contenido multimedia en la web sin tener que recurrir a *plugins* ni APIs propietarias (Villena, 2009).

Cascading Style Sheets 3: Constituye el estándar para la inserción de estilo (tamaños, colores, tipografías, espacios, bordes) a documentos estructurados, como, por ejemplo, documentos HTML y *Extensible Markup Language* (XML). El objetivo de la definición de este estándar es permitir la separación entre las normas de presentación y el propio contenido a mostrar (W3Schools, 2011).

PHP 5: Es un lenguaje de programación utilizado para la creación de sitios web. Es un lenguaje interpretado del lado servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. No necesita ser compilado para ejecutarse. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. PHP está diseñado específicamente para ser un lenguaje más seguro, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución y siguiendo algunas prácticas correctas de programación. Es multiplataforma y puede ser utilizado sobre los sistemas operativos: GNU/Linux, Windows, entre otros (Luca, 2004).

JavaScript 1.8.5:

JavaScript es un lenguaje utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes orientados a objetos mucho más complejos. Con JavaScript se pueden crear diferentes efectos e interactuar con usuarios (Valdés, 2007).

Este lenguaje posee varias características, entre ellas se destaca que es un lenguaje basado en acciones que posee menos restricciones. Además, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros (Valdés, 2007).

Java 8.0: Es un lenguaje muy valorado porque los programas desarrollados sobre se pueden ejecutar en diversas plataformas con sistemas operativos como Windows, Mac OS, Android, Linux o Solaris. Los programas Java son portables, es decir, independientes de la plataforma, porque pueden ejecutarse en cualquier ordenador o dispositivo móvil, independientemente del Sistema operativo que tengan instalado: Un programa Java puede ejecutarse en un ordenador de mesa, un ordenador portátil, una tableta, un teléfono, un reproductor de música o en cualquier otro dispositivo móvil con cualquier Sistema operativo (Guevara, 2012).

Lenguaje de modelado 8.0: Lenguaje de Modelado Unificado (*UML* por sus siglas en inglés). Está respaldado por el *Object Management Group* (por sus siglas en inglés *OMG*). El UML prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan (Popkin Software and Systems, 2007).

UML ayuda a especificar, visualizar y documentar los modelos de sistemas de *software*, incluyendo su estructura y diseño, de manera que cumpla con todos estos requisitos. Se puede usar UML para el modelado de negocios. Si se utiliza cualquiera de la gran cantidad de diagramas UML, se pueden analizar los requisitos de su aplicación futura y diseñar una solución que les satisface (OMG, 2015).

Se puede aplicar en el desarrollo de *software* entregando gran variedad de formas para dar soporte a una metodología de desarrollo de *software* (tal como el Proceso Unificado de Desarrollo o RUP), pero no especifica en sí mismo qué metodología o proceso usar. UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas, diagramas que son necesarios para la realización del módulo por lo que se selecciona como lenguaje de modelado a utilizar (Popkin Software and Systems, 2007).

1.2.2 Entornos Desarrollo Integrados

NetBeans: Esta es una herramienta de código abierto bajo licencia pública (GPL) que permite el desarrollo de aplicaciones con características modulares. Entre sus funcionalidades permite escribir, depurar, compilar y ejecutar programas, además soporta otros lenguajes de programación; es un producto libre y gratuito sin restricciones de uso, se pueden encontrar varias extensiones para el mismo. Entre las ventajas se puede destacar el auto-completamiento de código, interfaz para el diseño de GUI, la capacidad de importar clases, *software* multiplataforma y extensible. Entre los lenguajes de programación para los que brinda soporte se encuentra PHP, lenguaje sobre el que se desarrollan los módulos para Drupal 7 (NetBeans Community, 2000).

Aptana Studio: Es un entorno de desarrollo de código abierto basado en Eclipse que es otro entorno muy utilizado actualmente. Es una herramienta multiplataforma y está dirigida principalmente al desarrollo web con soporte para varios lenguajes de programación tal es el caso de HTML, CSS, JavaScript, PHP y otros. Proporciona la visualización de errores, presenta soporte para hacer FTP en servidores remotos. Contiene funcionalidades de edición, depuración, sincronización y administración de proyectos (Álvarez, 2011).

Eclipse: Es un entorno de desarrollo integrado, de Código abierto y Multiplataforma. Mayoritariamente se utiliza para desarrollar lo que se conoce como "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Es una potente y completa plataforma de Programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o

aplicaciones Java. No es más que un entorno de desarrollo integrado (IDE) en el que encontrarás todas las herramientas y funciones necesarias para tu trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar (The Eclipse Foundation, 2016).

Selección del Entornos Desarrollo Integrados

Se seleccionó NetBeans en su versión 8.1 para el desarrollo de la aplicación dado el gran soporte que brinda. Soporta varios lenguajes, tales como: PHP y todos los relacionados con la web (HTML, CSS, JavaScript y XML). Se caracteriza por su portabilidad ya que es compatible con la mayoría de los sistemas operativos como Windows, Mac, GNU/Linux y Solaris. De modo que se convierte en una herramienta muy útil para el desarrollo web.

Para el desarrollo de la aplicación se decide utilizar Eclipse ya que es de código abierto y multiplataforma permitiendo un mayor alcance de las aplicaciones desarrolladas sobre él. Este entorno de desarrollo integrado permite actualizarlo mediante *plugin* lo cual trae como ventajas que no será necesario el cambio de IDE para actualizar la aplicación solo la actualización del mismo.

1.2.3 Sistemas gestores de base de datos

DataBase Management System o Sistema de Gestión de Base de Datos (SGBD), es un tipo de *software* muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta (Rios, 2009). Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Entre los principales gestores de Bases de Datos a nivel mundial se encuentran Oracle, SQL Server y MySQL, PostgreSQL.

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. 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 (Gilfillian, 2010).

“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” (Miani, 2011).

SQLite: Es una herramienta de *software* libre, que permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser un teléfono celular. Implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, puede ser usado tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente (Montiel, 2008).

PostgreSQL: Es un sistema de gestión de base de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD. Presenta instalación ilimitada, es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento. Es multiplataforma, está disponible en casi cualquier Sistema Unix (34 plataformas en la última versión estable), y una versión nativa de Windows. Diseñado para ambientes de alto volumen, PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes (Miani, 2011).

Selección de Sistema gestor de BD

Los módulos de Drupal son independientes del sistema gestor de base de datos que se utilice ya que deben ser compatible con todos los que implementen lenguaje SQL. Para el desarrollo del Módulo de empaquetado de contenido web para Drupal 7 se decide optar por PostgreSQL en la versión 5.6.24 ya que este gestor es el que trae instalado por defecto el CMSDrupal 7.42, además esta versión incorpora avanzadas características en los temas de seguridad, soporte de aplicaciones, control y seguimiento, así como rendimiento y almacenamiento de datos especializados. También se decide usar de SQLite para la

aplicación en android dada su compatibilidad con este Sistema operativo y su rapidez en equipos con pocas capacidades de hardware como dispositivos móviles.

1.2.4 Servidor web

Nginx (“*Engine x*”) es un servidor HTTP y proxy inverso de alto rendimiento, y un servidor proxy para IMAP/POP3/SMTP. Es conocido por su estabilidad, gran conjunto de características, configuración simple, y bajo consumo de recursos. A diferencia de los servidores tradicionales no se basa en los hilos para manejar las peticiones. En su lugar, utiliza una arquitectura mucho más escalable orientada a eventos (asíncrono). Es muy útil al manejar miles de peticiones simultáneas dado su alto rendimiento y poco consumo de memoria (Inc Nginx, 2011).

Apache: Es un servidor web de código abierto, altamente configurable y modular. Utiliza Perl, PHP y otros lenguajes *scripts*. Soporta varios sistemas operativos: Linux, Solaris, Mac OS y Windows. Su función principal es analizar cualquier archivo solicitado por un navegador y mostrar resultados correctos de acuerdo con el código del archivo. Permite configurar los informes de errores, presenta visualización de códigos en numerosos niveles de HTML y la capacidad de determinar qué nivel del navegador puede aceptar el contenido. Es compatible con CGI¹, permite la personalización de variables y tiene soporte para reparación de errores. Es uno de los primeros servidores en soportar host basados en direcciones IP y host virtuales. Tiene un elaborado índice de directorios, un directorio de alias, informe de errores HTTP configurable, gestión de recursos para procesos hijos, reescritura de las URL, comprobación de ortografía de las URL y manuales *online* (Naramore, 2005).

Selección del servidor web

Los módulos de Drupal 7 son funcionales independiente del servidor web en que esté alojado el sitio web, para el desarrollo de esta solución se selecciona el servidor web Apache versión 3.0.2 porque no requiere la utilización de muchos recursos y permite que muchos lenguajes puedan ser utilizados del lado del servidor. Posee la capacidad de permitir la protección por contraseñas de las páginas de un gran número

¹ Interfaz de entrada común (en inglés *Common Gateway Interface*).

de usuarios, la visualización del código HTML en varios niveles, registra los errores en varios formatos y tiene soporte para *host* virtuales.

1.2.5 Herramientas de desarrollo

Content Management Systems o Sistemas de Gestión de Contenidos (CMS), son sistemas que permiten gestionar contenidos de un sitio web. Dicho de otra forma, un CMS es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web (Álvarez, 2011).

Los CMS son herramientas que cubren el ciclo de vida de las páginas de un sitio, permitiendo manejar su estructura, en aspectos relevantes como la creación de contenidos, actualización, administración de la información y la navegación de los usuarios (Robertson, 2008). Facilitan el trabajo de los desarrolladores; detrás de la interfaz de administración se trabaja con una gran base de datos donde se almacenan los contenidos y se actualizan a medida que se hacen los cambios deseados.

Drupal 7.41: Es un CMS modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Se destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web y un énfasis especial en la usabilidad y consistencia de todo el sistema. El diseño es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar diferentes tipos de sitios web. Dicho CMS presenta potencialidades que hacen de éste uno de los más utilizados a nivel mundial: Cuenta con una amplia comunidad de usuarios, los cuales se mantienen implementando nuevas versiones y módulos para lograr el crecimiento del CMS (Drupal, 2013). Además, este CMS presenta gran aceptación y uso dentro de Cuba para el desarrollo web

Presenta carácter modular basado en la inclusión de módulos elaborados por la comunidad de desarrolladores que proporcionan las más disímiles funcionalidades. Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario. El sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido. Los temas o plantillas de Drupal separan el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y PHP (Drupal, 2013).

Herramienta CASE

Visual Paradigm en su versión 8.0 es una herramienta CASE profesional que soporta la última versión de UML 8.0 así como el ciclo de vida completo del desarrollo de *software*, análisis y diseño orientados a objetos, construcción, pruebas y despliegue, exportación desde *Rational Rose*, exportación/importación XML, generación de informes y edición de figuras (Visual Paradigm, 2014).

Ventajas:

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse con la herramienta Netbeans IDE.
- Disponibilidad en múltiples plataformas.
- Ingeniería Inversa tiene capacidad de soporte para PHP.

PgAdmin III 3.0: Es una aplicación gráfica para el trabajo con el gestor de bases de datos PostgreSQL, de código abierto. Es capaz de gestionar versiones a partir de la versión PostgreSQL 7.3. Está escrita en C++ usando la librería gráfica multiplataforma wx Widgets, lo que permite que se pueda usar en Linux, y Windows.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP² y se puede encriptar mediante SSL para mayor seguridad (PgAdmin III, 2008).

1.2.6 Metodología de desarrollo de *software*

² Denominación común que se utiliza para designar al conjunto de protocolos de red en los que está basada Internet.

Como su nombre indica las metodologías son un conjunto de métodos, reglas, que por una parte sirven de guía para realizar los trabajos que van dando forma a los desarrollos y que por otra obligan a la dirección del proyecto y a los componentes de los equipos a realizar ciertas comprobaciones sistemáticas de modo que el resultado final, al menos desde un punto de vista formal, no presente incoherencias y esté dirigido a un objetivo claro y prefijado (Pressman, 2003).

La comparación o clasificación de metodologías no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalle, información disponible y alcance de cada una de ellas. A grandes rasgos, si se toma como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño, clasifican las metodologías en dos grupos: Metodologías Estructuradas y Metodologías Orientadas a Objetos. Por otra parte, considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales (o peyorativamente denominada Metodologías Pesadas, o Peso Pesado). Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso. A continuación, se revisan brevemente algunas de las metodologías más utilizadas.

OpenUP

Open Unified Process o Proceso Unificado Abierto es un modelo de desarrollo de *software*, desarrollado por la fundación Eclipse. Es un método y un proceso de desarrollo de *software*, apropiado para proyectos pequeños y de bajos recursos; es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo. Permite detectar errores tempranos a través de un ciclo iterativo. Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP. Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas. El ciclo de vida de OpenUp consiste de cuatro fases: Concepción, Elaboración, Construcción y Transición (Tusa, 2012).

XP

Extreme Programming es una metodología ágil creada por Kent Beck en 1997 para la plantilla del proyecto C3 en Chrysler. Es diseñada para entornos dinámicos en los cuales se presentan equipos de desarrollo

pequeños (hasta 10 programadores). Orientada fuertemente hacia la codificación y programación de esta forma disminuyendo el tiempo de desarrollo de los proyectos. Crea énfasis en la comunicación informal, verbal con muy poca documentación necesitada en el desarrollo del sistema. La planificación de la misma es por entregas contando con las siguientes características: (Letelier, 2005)

- Planificación por entregas (*releases*).
- Se priorizan aquellas *user-story*³ que el cliente selecciona porque son más importantes para el negocio.
- Entregas:
 - ❖ Son lo más pequeñas posibles.
 - ❖ Se dividen en iteraciones (iteración = 2 o 3 semanas).
 - ❖ Están compuestas por historias.
- A cada programador se le asigna una tarea de la *user-story*.
- La programación de tareas se realiza por parejas.
- La pareja diseña, prueba, implementa e integra el código de la tarea.
- Código dirigido por las pruebas.
- Código modular, intentando refactorizar siempre que se pueda.

Metodología AUP

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de Desarrollo (RUP). Este describe de manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles incluyendo (Flores, 2012):

³ Historias del Usuario establecen los requisitos del cliente que no son más que trozos de funcionalidad que aportan valor definido por el cliente.

- Desarrollo dirigido por pruebas (*test driven development*).
- Modelo Ágil.
- Gestión de Cambios Ágil.
- Refactorización de Bases de Datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva.

Selección de la metodología

Para el desarrollo del sistema se selecciona la metodología AUP ya que existen adaptaciones para el centro en el cual se desarrollará la aplicación. Esta metodología además es ágil por lo que es compatible con equipos de trabajos pequeños como en el caso de este proyecto. Esta metodología además es ágil por lo que es compatible con equipos de trabajos pequeños como en el caso de este proyecto y tiene un enfoque centrado al cliente con iteraciones cortas lo cual permite la detección de errores tempranos.

1.2.7 Herramientas para pruebas del sistema

En la realización de las pruebas de rendimiento que se realizaron al Módulo de empaquetado web de contenidos para Drupal 7 se decidió hacer uso de la herramienta JMeter en su versión 2.12.

JMeter: Es un *software* de código abierto, diseñado para pruebas de carga de comportamiento funcionales y la medición del rendimiento. Originalmente fue diseñado para probar las aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba. Es utilizado para probar el rendimiento tanto en los recursos estáticos como dinámicos. Puede ser utilizado para simular una carga pesada en un servidor, de red o un objeto para poner a prueba su resistencia o para analizar el rendimiento general en diferentes tipos de carga. Puede usarse además para hacer un análisis gráfico de rendimiento o para probar el comportamiento de su servidor / script / objeto con carga pesada concurrentes (Foundation, 2011).

Acunetix: Es un escáner de vulnerabilidades en servidores web. Es capaz de detectar vulnerabilidades como inyección de SQL, XSS, XXE, FRSS, ataques de encabezado de host y más de 3000 otras vulnerabilidades web (Acunetix, 2016).

1.3 Conclusiones parciales

El estudio realizado a los sistemas homólogos permitió identificar que ninguno de los sistemas le daba solución al problema, por lo que quedó demostrado la necesidad de implementar como solución informática una combinación de un módulo para Drupal y una aplicación para Android. Además, el estudio de las herramientas, tecnologías y metodologías permitió definir la base tecnológica para el desarrollo de la propuesta de solución. Luego de realizar un estudio de las tecnologías y herramientas se determinó que se puede realizar el desarrollo del módulo haciendo uso de cualquier Gestor de Base de Datos y Servidor Web por las facilidades que brinda el CMS Drupal.

Capítulo 2. Características del Módulo de empaquetado web de contenido para Drupal 7

2.1 Introducción

En el presente capítulo se describe el entorno mediante un modelo de dominio en el cual se analizan conceptos, entidades y sus relaciones, que están relacionados con el entorno. También se definen las principales características que debe cumplir el sistema en términos de requisitos funcionales y no funcionales. Así como se presentan los diagramas de clases de diseño para una visión general del funcionamiento de la aplicación y se describen las historias de usuarios que se materializarán en funcionalidades.

2.2 Modelo de dominio

El modelo del dominio muestra clases conceptuales significativas en un dominio de problema; es un artefacto de la disciplina de análisis construido durante la fase de inicio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de *software* sino de la propia realidad física (Larman, 2003).

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes *software*. No se trata de un conjunto de diagramas que describen clases *software*, u objetos *software* con responsabilidades (Larman, 2003).

Se puede concluir que los modelos de dominio se utilizan para la modelación de los procesos que intervienen en el negocio, en el cual se representan los conceptos más importantes y significativos en el entorno del problema a resolver, pudiendo ayudar en parte como un punto de partida para el diseño de este. Similares a los mapas mentales utilizados en el aprendizaje constituye un medio para comprender el sector de negocios al cual el sistema va a servir. Además, permite identificar las interrelaciones entre las entidades o clases comprendidas en el ámbito del dominio del problema, al igual que sus atributos y restricciones.

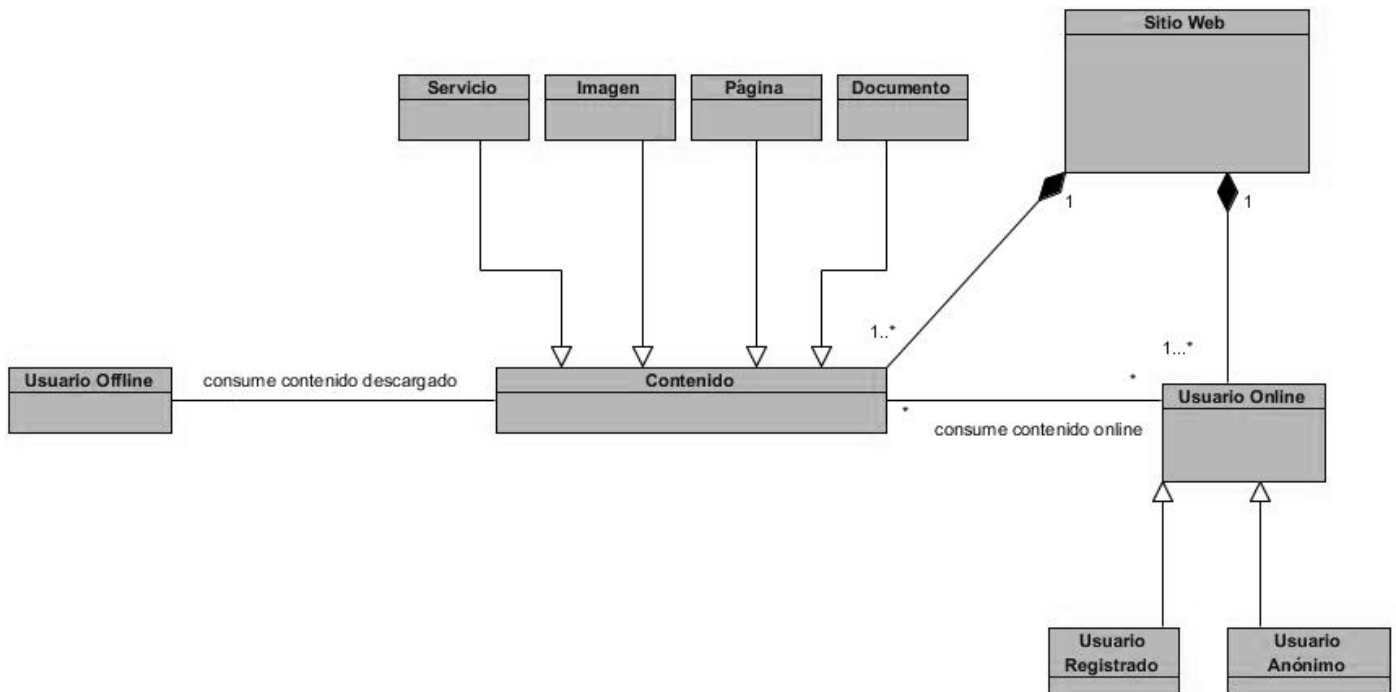


Figura 1. Diagrama de Clases del Modelo de Dominio.

1.2.1 Descripción de las clases del Modelo de Dominio

A continuación, se explica en qué consisten cada una de las clases que conforman el modelo de dominio.

Sitio Web: Conjunto de componentes que brinda información y/o servicios.

Usuarios Online: Personas que interactúan con el portal web mediante la red.

Usuarios Anónimo: Se refiere al usuario que no presenta una identificación previa.

Usuarios Registrado: Es el usuario que tiene ciertos privilegios para interactuar con las funcionalidades del sitio web, pero no con la administración del sitio web, presenta una identificación previa.

Servicios: Conjunto de funcionalidades con el objetivo de responder a las necesidades de los usuarios y de la organización.

Imagen: Representación visual de objetos o formas de manera digital.

Página: Es un documento electrónico el cual contiene información textual, visual y/o sonora que se encuentra alojado en un servidor y puede ser accesible mediante el uso de navegadores.

Documento: Se refiere a elementos de hipertexto presente en las páginas.

Contenido: Combinación de los elementos página, imagen, documento y servicio.

Usuario *Offline*: Es aquel usuario que luego de descargar un contenido del sitio puede acceder a este sin conexión.

2.3 Levantamiento de requisitos

El proceso de captura de requisitos inicia con la interacción con el cliente. Durante este proceso se hace una relación de las necesidades que tiene el usuario y definir lo que debe hacer el sistema. Se describen las condiciones que se necesitan para que los requisitos funcionales puedan ser cumplidos o como se conocen comúnmente los requisitos no funcionales. Se detallan los usuarios que van a interactuar con el sistema y los niveles de acceso o permisos que tendrán en el sistema.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación (Turner, 2005).

2.3.1 Requisitos Funcionales y Requisitos no Funcionales

La tarea de asignar prioridades requiere de la participación de clientes y usuarios con cierto nivel de decisión y puede realizarse de diversas maneras, tales como reuniones, cuestionarios y otras. Se debe determinar la importancia relativa que tiene un requisito para los clientes y usuarios, y organizar aquellos requisitos que deben implementarse inicialmente frente a aquellos que pueden postergarse. Al asignar prioridades, se deben tener en cuenta la dependencia entre requisitos, la multiplicidad de intereses de los clientes y usuarios, las limitaciones de recursos, las necesidades del negocio, las imposiciones del mercado y los costos de implementación, entre otros factores.

En el caso de la investigación desarrollada para definir la prioridad de los requisitos funcionales se etiquetó como prioridad media aquellos requisitos que no resultan imprescindibles para el correcto funcionamiento

del sistema. y prioridad alta los que representan la columna vertebral para el funcionamiento del módulo de empaquetado web de contenido.

Tabla 1. Requisitos Funcionales.

Código	Descripción (Requisitos Funcionales) del Módulo	Prioridad
RF1	Seleccionar Tipo de Contenido.	Alta
RF2	Seleccionar Elementos del Menú.	Alta
RF3	Seleccionar Campos por Tipo de Contenido.	Alta
RF4	Enlazar Tipo de Contenido con Menú.	Alta
RF5	Generar Base de Datos para APK.	Alta
RF6	Actualizar Base de Datos en Portal.	Alta
Código	Descripción (Requisitos Funcionales) de la Aplicación	Prioridad
RF7	Consumir Base de Datos generada por el módulo.	Alta
RF8	Mostrar listado de contenidos.	Media
RF9	Mostrar contenido.	Alta
RF10	Mostrar menú de navegación.	Media
RnF (Requisitos no Funcionales) para el Módulo		
	Usabilidad	
RnF-1	El sistema debe garantizar una interfaz amigable e intuitiva para asegurar la fácil navegación del usuario por la misma.	
	Confiabilidad	
RnF-2	Se asignarán los permisos de acceso en dependencia del rol que desempeñe cada usuario del sistema.	
	Eficiencia	

RnF-3	El sistema debe ser capaz de procesar 10 peticiones por segundo para un total de 20 usuarios concurrentes.
RnF-4	El sistema debe tener un tiempo de respuesta inferior a 1 minuto.
	Restricciones del Diseño
RnF-5	Las páginas generadas deben cumplir con los estándares para XHTML 2.0 y CSS 3.0.
RnF-6	El software, bibliotecas o componentes empleados deben estar regidos en la medida de lo posible por licencias y patentes de software libre.
	Interfaz
RnF-7	La aplicación debe ser visible en navegadores como Mozilla Firefox versión superior a la 30, Chrome versión superior a la 31 y Opera 11 o superior.
RnF-8	La comunicación entre el cliente y el servidor web será realizada a través del protocolo HTTPS (<i>Hypertext Transfer Protocol Secure</i> , siendo en español: Protocolo seguro de transferencia de hipertexto).
	Requisitos de licencia
RnF-9	Se debe utilizar la licencia PHP License.
RnF-10	Se debe utilizar la licencia BSD de PostgreSQL.
RnF-11	Se debe utilizar la licencia GNU/GPL para el CMS Drupal.
	Estándares aplicables
RnF-12	Se siguen el estándar de programación sugerido por los desarrolladores del CMS Drupal.
	Seguridad
RnF-13	Se garantizará la integridad y confidencialidad de la información mediante mecanismos de control de accesos no autorizados utilizando: usuario, contraseña y niveles de accesos para cada usuario, de manera que cada uno pueda tener disponible solamente las opciones relacionadas con su actividad y el acceso a datos propios.
RnF-14	Las contraseñas en la base datos estarán encriptadas por DM5.

RnF (Requisitos no Funcionales) para la aplicación	
	Usabilidad
RnF-15	La aplicación debe de ser fácil de descargar e instalar.
RnF-16	Las interfaces de la aplicación deben presentar una vista limpia y de fácil navegación.
	Eficiencia
RnF-17	El tiempo de respuesta de la aplicación debe de ser inferior a 3 segundos.
	Hardware
RnF-18	La aplicación necesita al menos 2 megas de almacenamiento para ser instalada y funcione correctamente.
RnF-19	La aplicación requiere como mínimo sistemas operativos superiores a Android 4.1.
	Seguridad
RnF-20	Debe mantener los datos almacenados seguros permitiendo solo lectura de los mismos.
	Estándares aplicables
RnF-21	Se siguen el estándar de programación sugerido por los desarrolladores de Java.

2.3.2 Historias de Usuario

Las HU sirven para registrar los requerimientos de los clientes según el negocio y son utilizadas para poder realizar la estimación de cada una de las iteraciones durante la fase de planificación. Las HU son escritas por el equipo de trabajo en conjunto con los clientes en base a lo que se estima que es necesario para el sistema. Están escritas en un formato de oraciones en la terminología del cliente, sin necesidad de sintaxis técnicas. También son utilizadas para poder crear las pruebas de aceptación. Las HU solo proveen suficiente detalle para poder realizar la estimación de cuánto tardará en ser implementada dicha funcionalidad. Una gran diferencia entre las HU y los documentos tradicionales es que se centran en lo que el cliente necesita (BALAREZO, 2013).

La prioridad en el negocio:

- **Alta:** Cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
- **Media:** Cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
- **Baja:** Cuando constituyen procesos que se deben tener en cuenta, pero su ausencia no perjudica el flujo principal del negocio.

El riesgo en desarrollo:

- **Alto:** Cuando en la implementación de las HU pueden surgir errores que lleven a la inoperatividad del código.
- **Medio:** Cuando en la implementación de las HU pueden existir errores que retrasen la entrega del producto.
- **Bajo:** Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Un Punto de Estimación equivale a una semana de programación, una semana de programación corresponde a 40 horas en desarrollo, 8 horas durante 5 días de la semana del calendario normal.

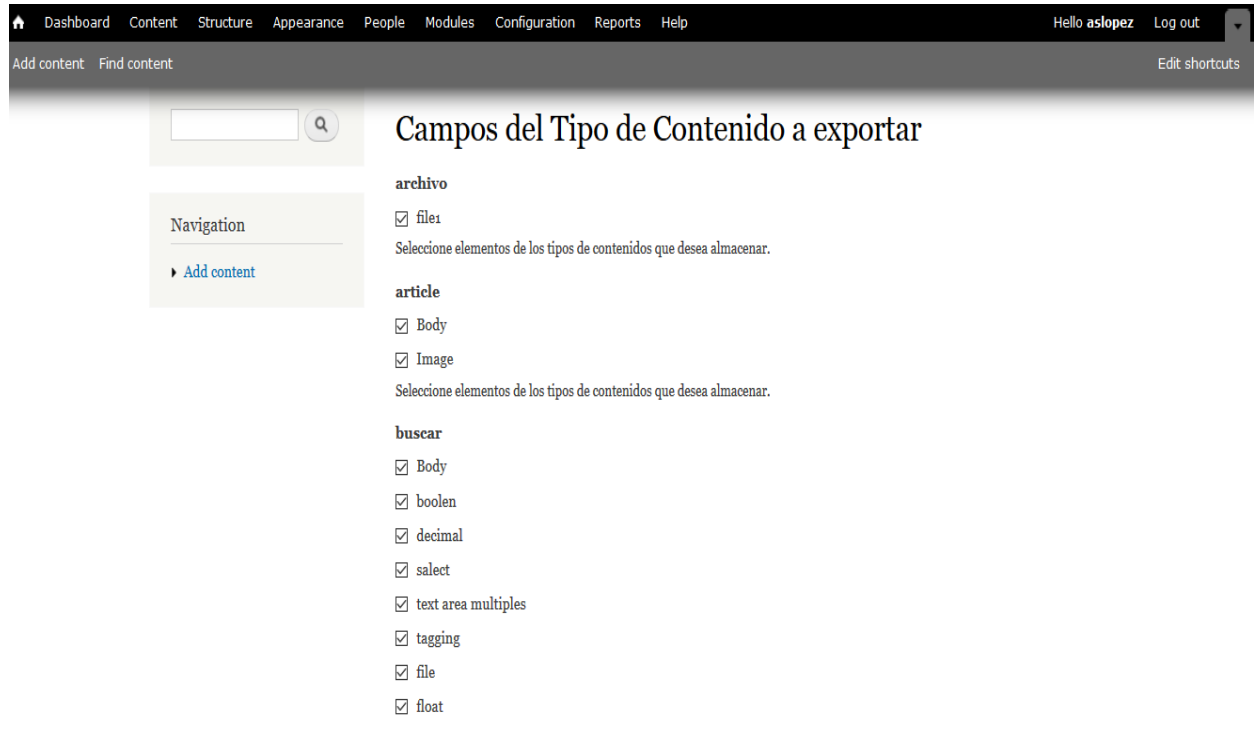
A continuación, la HU del sistema perteneciente al RF-8. Seleccionar Campos por Tipo de Contenido.

Tabla 2. Descripción de HU: Seleccionar Campos por Tipo de Contenido.

Historia de usuario	
Número: HU_8	Nombre Historia de Usuario: Seleccionar Campos por Tipo de Contenido
Programador: Abel Salas López	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 1 días
Riesgo en Desarrollo: Alta	Tiempo Real: 1
Descripción: Muestra un listado con los campos correspondientes a cada tipo de contenido, los que pueden ser seleccionados por los usuarios autenticados para ser almacenar en la base de datos.	

Observaciones: El tipo de contenido no puede quedar sin ningún campo y el campo título del tipo de contenido es seleccionado por defecto.

Prototipo de interfaz:



2.5 Arquitectura

El sistema implementado sobre el CMS Drupal hereda la arquitectura de este. El mismo para permitir una mayor flexibilidad y facilidad en la creación de sitios web utiliza la abstracción por capas que aplica en el tratamiento de contenido. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una serie de elementos básicos. Estos son los nodos (*nodes*), módulos (*modules*), bloques y menús (*blocks & menus*), permisos de usuario y plantillas (*templates*). La presente propuesta de solución presenta una arquitectura en capas (Figura 2).

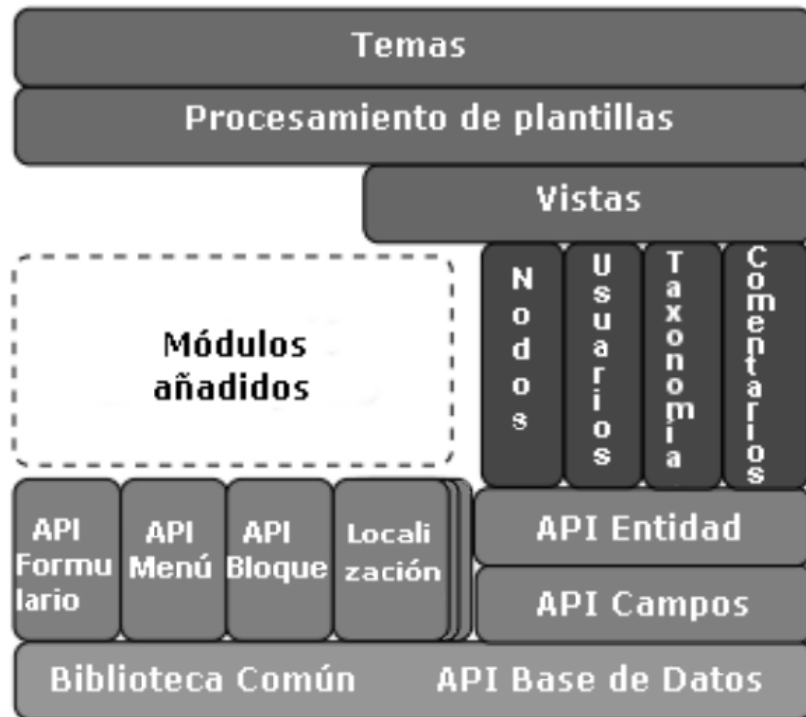


Figura 2. Arquitectura de Drupal (SeeD, 2015).

Nodos (Nodes): Son los elementos básicos en que Drupal almacena la información, los contenidos. Así a medida que el sitio web crece, lo va haciendo el número de *Nodes* los cuales van formando un “depósito de *Nodes*” cada vez mayor. Se puede decir que la primera capa de la estructura de Drupal la forma este “depósito” de *Nodes*. En el desarrollo del Módulo de empaquetado web de contenido para Drupal 7 se decide trabajar con los nodos almacenados en la base de datos, permitiendo al usuario definir que nodos desea obtener y exportar.

Módulos (Modules): Es la capa encargada de mostrar en los temas los cambios realizados a través de los módulos. Se complementan con las entidades y campos ampliando las posibilidades de presentación de contenidos dinámicos. En esta capa se encuentra el módulo desarrollado permitiendo ver los cambios en los temas.

Entidades (*Entity*): Las entidades son tipos de objetos a los que se le pueden asociar campos, alguna de estas es: nodos, usuarios, taxonomía, archivos. En el caso del módulo a desarrollar no presenta nuevas entidades, sino que consume las presentes en el sitio para ser exportadas según los criterios de selección del usuario.

Plantillas (*Templates*): Establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del sitio web sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que se quiere es renovar la apariencia de un sitio web.

Vistas (*View*): Permite definir diversas formas de visualización de la información dentro del sitio desarrollado. Se evidencia en las diversas vistas presentadas por el módulo durante el proceso de generación de la base de datos.

Base de datos: Esta es la capa de abstracción de la base de datos que permite la comunicación del gestor de contenido con el gestor de base de datos, permitiendo mostrar contenidos en las vistas del sistema.

2.5.1 Patrones de Diseño

En el desarrollo de software el uso de patrones de diseño constituye una buena práctica, proporcionando una organización estructurada entendible por cualquier programador. El uso de patrones permite construir software que sean más fáciles de mantener y extender, representando un ahorro de tiempo (Gama y autores, 2003). A continuación, se describen los principales patrones empleados en el desarrollo de la propuesta de solución:

Observer (Observador): El patrón observador es generalizado en Drupal, evidenciándose en los módulos, que implementan *hooks* (ganchos) determinados para eventos de inserción o actualización de una determinada entidad, siendo declarados como observadores de las entidades con las que interactúan. Se puede ver presente en la implementación del *hook forms_form2_validate*, este espera una petición de *submit* para capturar los datos y verificar que sean correctos de lo contrario lanza un error.

Chain of Responsibility (Cadena de Responsabilidad): El sistema de menú de Drupal sigue el patrón Cadena de Responsabilidad. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y que función se llama para

ejecutar la petición. Si se puede ejecutar la acción, se devuelve la información solicitada, si no, pasa a la siguiente comprobación de la solicitud hasta que un módulo atienda la petición o hasta que un módulo deniegue el acceso. Este patrón se puede ver evidenciado cuando se va acceder al módulo es necesario ir al menú de Configuración y luego Descarga_apk, estos dos menús se aseguran de verificar que usuario tenga acceso al contenido solicitado.

Bridge (puente): La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón puente. Los módulos necesitan ser programados independientes del gestor de bases de datos que se esté usando. Este patrón es usado repetidamente en la solución desarrollada dada la necesidad de hacer varias consultas a la base de datos. Ejemplo de esto es la función `llenar_bd()` donde se hacen consultas de selección e inserción a la base de datos abstrayéndose del SGBD que se emplee para manejar los datos almacenados.

2.6 Diseño

2.6.1 Diagrama de Clases de Diseño

El Diagrama de Clases es uno de los diagramas principales de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.

Con el objetivo de alcanzar un mayor grado de comprensión de la aplicación a desarrollar, es indispensable tener en cuenta el funcionamiento del CMS Drupal, específicamente los paquetes que este presenta. En el siguiente diagrama se muestran los mismos, con su descripción correspondiente:

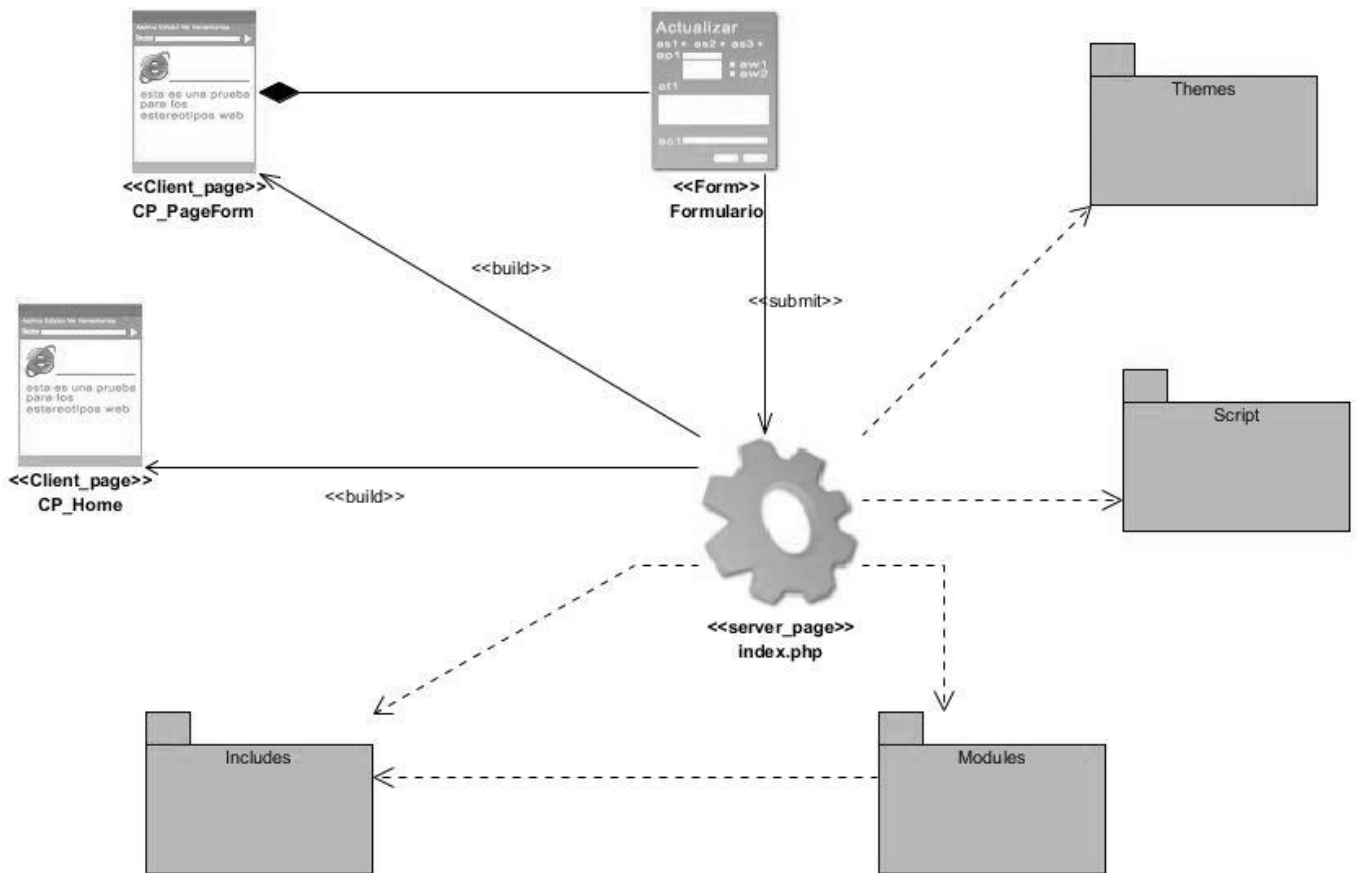


Figura 3. Diagrama de Clases de Diseño de Drupal.

Includes: En este paquete se encuentran los ficheros de conexión a la base de datos del sistema.

Modules: Espacio en el que se encuentran todos los módulos que forman parte del funcionamiento del CMS.

Themes: Agrupa las plantillas que forman parte del tema de la interfaz del sistema. Cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.

Scripts: Almacena un grupo de ficheros que contienen código para el correcto funcionamiento del sistema, permitiendo que se visualicen los datos correctamente.

Una vez analizado el diagrama de clases de diseño de la plataforma, se modelan los diagramas de clases de diseño para cada escenario como se muestra a continuación. Los restantes diagramas se ubican en el Anexo 3.

Seleccionar Campos por Tipo de Contenido

Entre las principales funcionalidades que debe tener implementada la solución se encuentra la de Seleccionar Campo por Tipo de contenido. A continuación, se describen los distintos escenarios:

En el caso de Seleccionar Campo por Tipo de Contenido la página servidora SP_Seleccionar_Campo_por_Tipo_de_Contenido construye la página cliente SP_Seleccionar_Campo_por_Tipo_de_Contenido, la cual se compone por el formulario Frm_Seleccionar_Campo_por_Tipo_de_Contenido que permite la selección de los Tipo de Contenidos que se van a almacenar en la base de datos los cuales son procesados por la página servidora.

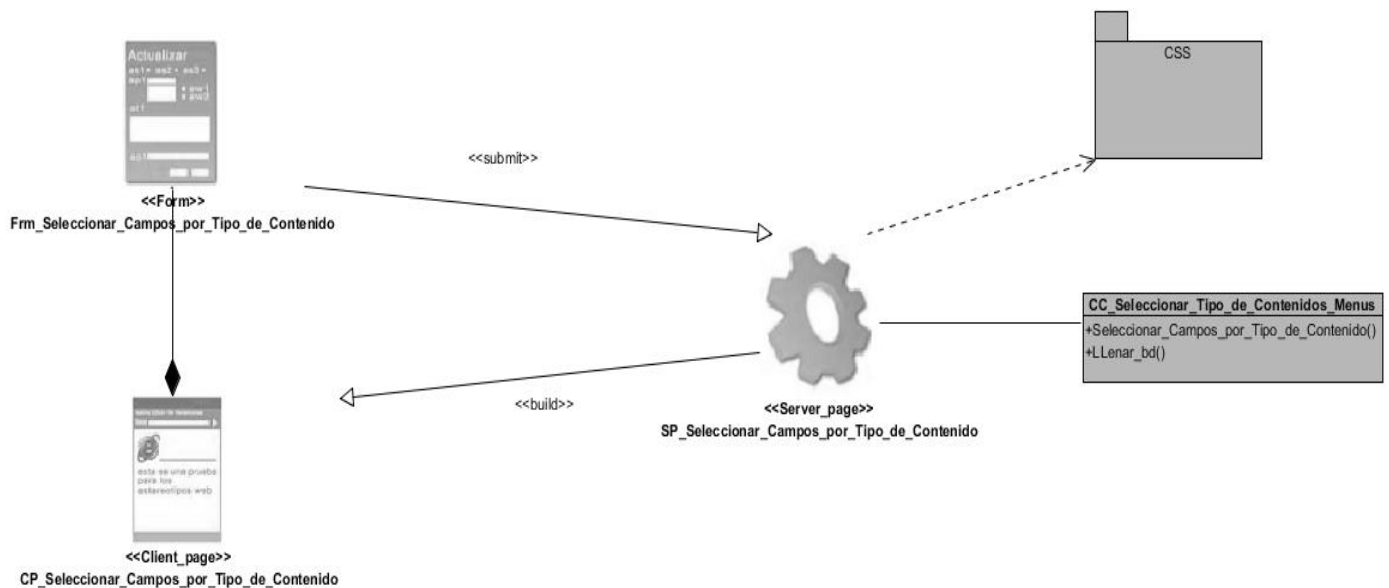


Figura 4. Diagrama de clases de diseño. Seleccionar Campo por Tipo de Contenido.

2.7 Modelo de datos

Mecanismo formal para representar y manipular información de manera general y sistemática, mediante la descripción de los datos, operaciones y las reglas de integridad. Dentro de estos existe un gran conjunto de

opciones como los modelos basados en grafos, multidimensional, orientados a objetos, relacional, entre otros. En la presente investigación se optó por un modelo relacional para la representación de los datos y sus relaciones entre sí.

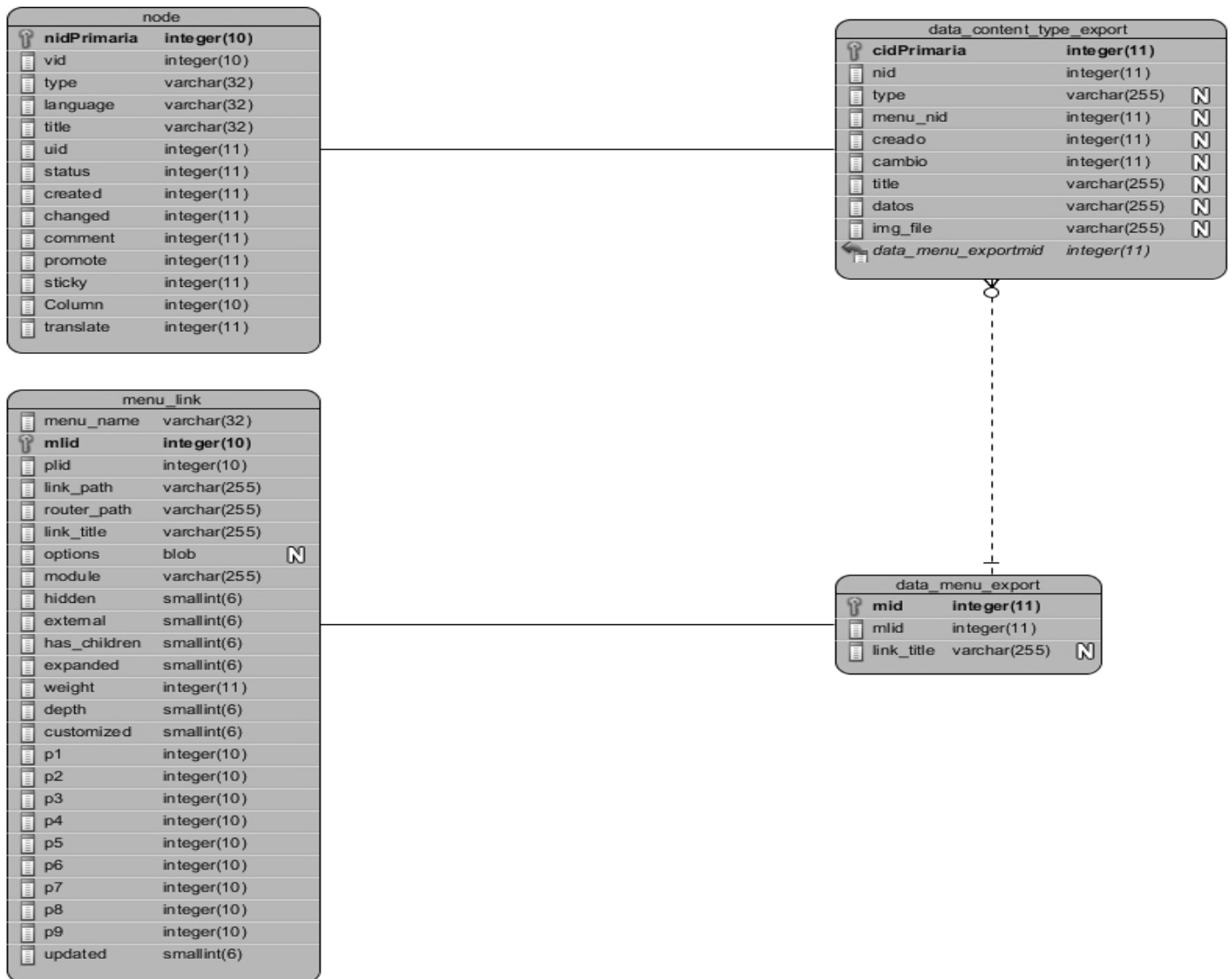


Figura 5. Modelo relacional

En el modelo relacional presentado se pueden observar dos tablas internas de Drupal, node y menu_link. Por otra parte, se presenta las tablas data_content_type_export y data_menú_export presentando una relación de uno a muchos ya que un menú puede tener varios tipos de contenidos. La tabla

data_content_type_export almacena un conjunto de datos extraídos de la tabla node como cid (id del contenido), nid (id del nodo), type (tipo de contenido), menú_nid (id del menú con quien se relaciona), creado (creación el nodo), cambio (última actualización del nodo), title (titulo), datos (todos los datos que sean texto) y img_file (url, id de las imágenes y documentos). La tabla data_menú_export presenta las columnas mid(id el menú), mlid (id del link del menú), link_title (título del link) las cuales obtienen su datos de la tabla de Drupal menú_link.

2.8 Modelo de despliegue

El diagrama de despliegue, es utilizado para representar de forma visual las relaciones físicas que existen entre los componentes de software y hardware en el sistema. El mismo representa nodos y las relaciones entre ellos, los nodos son conectados por enlaces de red, conexiones TCP/IP, HTTP o HTTPS.

En la figura se muestra el diagrama de despliegue correspondiente a la presente solución. El nodo “**Dispositivo cliente**” representa las computadoras o dispositivos de sistema Android de los usuarios que se conectan al sistema a través de un navegador web, las cuales realizan peticiones al Servidor Web (nodo “**Servidor Web**”) donde estará alojado el sistema mediante el protocolo HTTP. Este servidor mantendrá una conexión mediante la familia de protocolos TCP/IP con el servidor de bases de datos PostgreSQL o MySQL representado por el nodo “**Servidor de BD**”.



Figura 6. Diagrama de Despliegue.

2.8 Conclusiones parciales

Con la culminación de este capítulo se puede concluir que:

El Modelo de Dominio realizado a partir de los procesos identificados permitió conocer todos los términos y conceptos presentes en el entorno, los cuales fueron descritos y especificados representados en un diagrama de clases, que permitió comprender mejor el funcionamiento e interrelación de los mismos.

Durante el Levantamiento de Requisitos, se consiguió identificar las funcionalidades que el Módulo de empaquetado web de contenidos para Drupal 7 debe brindar y las restricciones sobre las que va a operar.

La conformación de los Diagramas de Clases del diseño permitió definir las relaciones entre las clases del software permitiendo visualizar la relación entre las mismas, así como las funcionalidades y atributos que deben presentar cada una de estas.

Con el modelo de despliegue se permitió modelar una vista de las topologías del hardware sobre el que se ejecuta el sistema.

Capítulo 3: Implementación y Prueba del Módulo de empaquetado web de contenido para Drupal 7

El presente capítulo describe la implementación y prueba del sistema, lo cual forma parte de la fase de construcción, mediante el diagrama de componentes se describe como los elementos del modelo de diseño se implementan en términos de componentes. Se realizan pruebas de caja negra al módulo y a la apk desarrollado para comprobar su funcionamiento, con el Objetivo de detectar y corregir las posibles no conformidades.

3.1 Diagrama de componentes

Los diagramas de componentes son utilizados para estructurar el modelo de la implementación. Permiten modelar una vista estática del sistema, muestran la organización y las dependencias lógicas entre un conjunto de componentes del software, que pueden ser bibliotecas, ejecutables y códigos fuentes. A continuación, se representa dicho diagrama para la solución que se desarrolla:

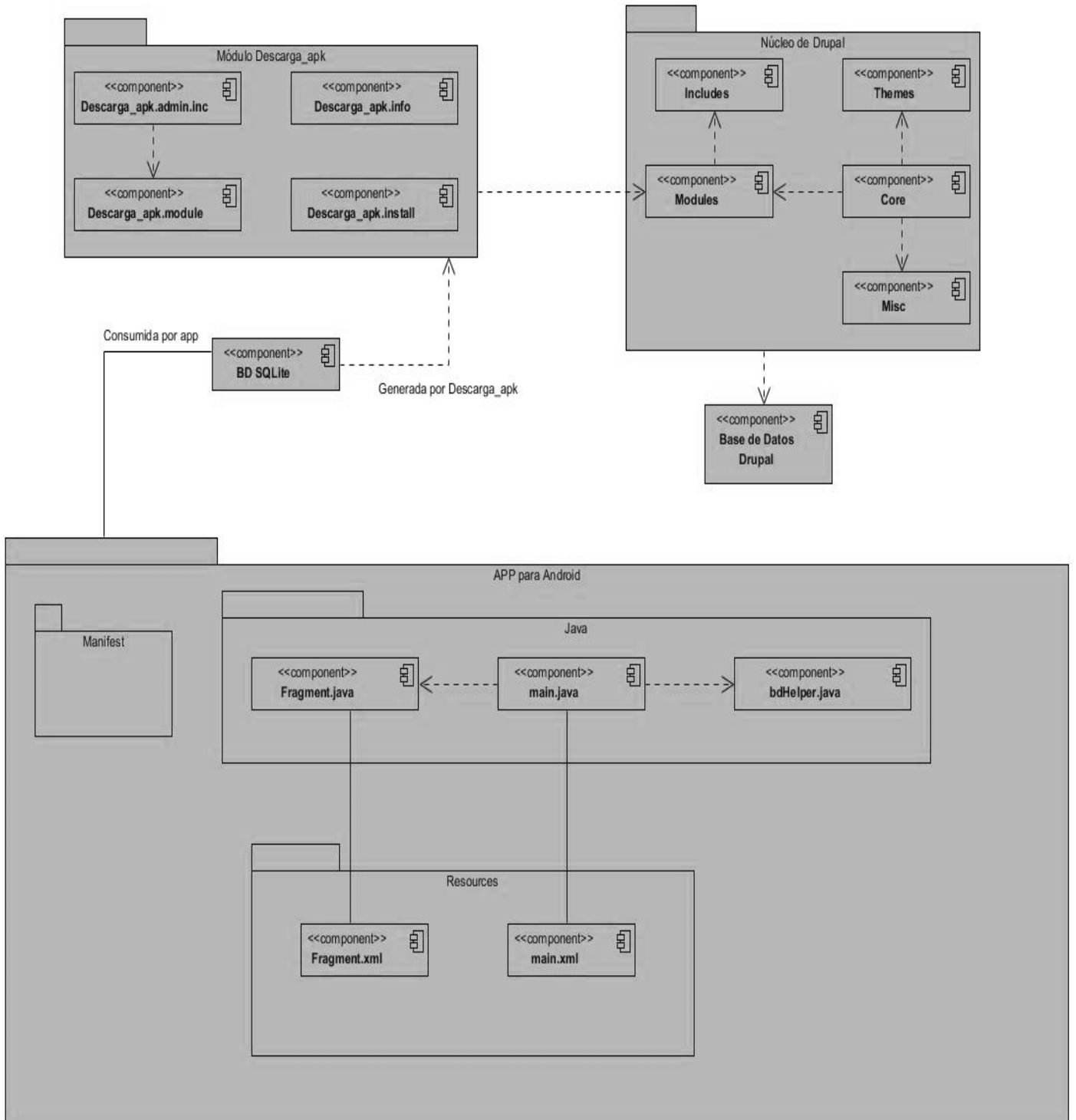


Figura 7. Diagrama de Componentes.

La tabla que se muestra a continuación describe cada uno de los componentes representados en el diagrama.

Tabla 3. Descripción de componentes

COMPONENTES	DESCRIPCIÓN
<i>Themes</i>	Se encuentra toda la capa de presentación y diseño.
<i>Core</i>	Es el punto de inicio del portal, a partir de esta entrada se invocan los diferentes módulos del CMS.
<i>Modules</i>	Directorio donde se encuentran los módulos de Drupal.
<i>Include</i>	Este componente contiene un conjunto de ficheros indispensables para el funcionamiento de Drupal, tales como funciones y scripts necesarios para el funcionamiento de los módulos y del sistema.
<i>Misc</i>	Incluye elementos que tienen que ver con el diseño, y funcionamiento (imágenes, ficheros js, etc).
Base de datos Drupal	Es la base de datos de Drupal.
Descarga_apk.info	Archivo que incluye la definición del módulo.
Descarga_apk.install	Define funciones a ejecutarse durante la instalación.
Descarga_apk.module	Contiene todas las funciones definidas en el módulo.
Descarga_apk.admin.inc	Página de administración del módulo.
BD SQLite	Es la base de datos generada por el módulo.
Main.java	Define las funciones principales de la aplicación.

Fragment.java	Define la estructura y funcionalidades de un fragmento para ser consumidos en main.java
bdHelper.java	Es el encargado de permitir la comunicación con la base de datos SQLite.
Main.xml	Representa la estructura de la vista principal del sistema.
Fragment.xml	Representa la estructura que representa a cada fragmento generado.

3.2 Estándares de codificación

Los estándares de código resultan importantes en cualquier proyecto de desarrollo, ya que determinan normas de estilos, orden y estructura del lenguaje. Los estándares de código ayudan a asegurar que el código tenga una alta calidad, menos errores, y pueda ser mantenido fácilmente.

A continuación, se exponen varios estándares que presenta Drupal (Buytaert, 2016), utilizados durante el desarrollo del sistema (ver **Tabla 4 Estándar de codificación Drupal**):

Tabla 4. Tabla Estándar de codificación Drupal

Estándar de codificación	Descripción
Indentación	Consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En Drupal la indentación es con 2 espacios, nunca con tabuladores y evitando dejar espacios en blanco al final de cada línea.
Etiquetas de apertura y cierre de PHP	Utilizar las etiquetas <code><?php</code> y <code>?></code> . En general se omite la etiqueta de cierre de PHP (<code>?></code>) al final de los archivos <code>.module</code> y <code>.inc</code> .

<p>Operadores</p>	<p>Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio.</p>
<p>Uso de comillas</p>	<p>Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto o cuando el texto puede incluir alguna comilla simple.</p>
<p>Estructuras de control</p>	<p>Debe existir un espacio entre el comando que define la estructura (<i>if</i>, <i>while</i>, <i>for</i>, etc.) y el paréntesis de apertura.</p> <ul style="list-style-type: none"> - La llave de apertura (<code>{</code>) se situará en la misma línea que la definición de la estructura, separada por un espacio. - Usar siempre las llaves <code>{}</code> aún en los casos en que no sea obligatorio su uso. - Las estructuras <i>else</i> y <i>elseif</i> se escribirán en la línea siguiente al cierre de la sentencia anterior.
<p>Funciones</p>	<p>Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Se debe incluir siempre como prefijo el nombre del módulo, tema, etc. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.</p>

Arreglos	Los valores dentro de un <i>array</i> (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados. Cuando la línea de declaración del <i>arrays</i> supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos.
Constantes	Los nombres de las constantes deben escribirse en mayúsculas, con guiones bajos para separar palabras.
Variables globales	Se declaran utilizando un guión bajo inicial, seguido del nombre del módulo o tema y otro guión bajo antes del nombre de la variable.
Nombres de módulos	El nombre de un módulo nunca debería incluir guiones bajos, aunque se componga de varias palabras.
Nombres de archivos	Los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación, que tendrás extensión .txt y el nombre es mayúsculas.

<p>URL de ejemplo</p>	<p>Por convención, siempre que haya que indicar una URL de ejemplo, se debe utilizar "<i>example.com</i>".</p>
------------------------------	--

En la siguiente tabla, se exponen varios estándares que presenta Java, utilizados durante el desarrollo del sistema (ver **Tabla 5 Estándar de codificación Java**):

Tabla 5. Estándar de codificación Java.

<p>Nomenclatura</p>	<p>El idioma por defecto para proveer sentido funcional al nombre de clases, variables y constantes, será una mezcla entre la nomenclatura tradicional en inglés y la nomenclatura funcional adoptada en este caso español.</p>
<p>Nombres de clases</p>	<p>Los nombres de clases deben ser mezclas de mayúsculas y minúsculas, con la primera letra de cada palabra interna en mayúsculas. Se debe mantener los nombres de clases simples y descriptivos.</p>
<p>Comentarios</p>	<p>Los comentarios serán utilizados para dar información adicional al desarrollador sobre la implementación del diseño de la clase. Por tanto, se debe evitar referencias al diseño funcional de la misma y el uso de caracteres especiales.</p>
<p>Indentación</p>	<p>Consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. Java utiliza cuatro espacios como unidad de indentación.</p>
<p>Longitud de la línea</p>	<p>Evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.</p>

Sentencias	Cada línea debe contener como mucho una sentencia.
Espacios en blanco	Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas.

3.3 Pruebas

Durante el flujo de trabajo de pruebas se verifica el resultado de la implementación probando la estructura, tanto en la construcción interna como intermedia, así como las versiones finales del sistema a ser entregado (Ivar G. B. Jacobson, 2000). Concretamente los objetivos de las pruebas son:

- Planificar las pruebas.
- Diseñar e implementar las pruebas.
- Aplicar las pruebas y manejar los resultados.

Su principal objetivo es evaluar o valorar la calidad del producto. Las pruebas constituyen un elemento crítico para la garantía de la calidad del software (Ivar G. B. Jacobson, 2000).

3.3.1 Funcionales

Para la realización de estas pruebas se utilizó la técnica de caja negra las cuales son realizadas desde el punto de vista de la interfaz de usuario, lo que permitió valorar el funcionamiento de la aplicación según la interacción del usuario con el portal.

A continuación, se describen los casos de prueba para algunos requisitos con prioridad Alta:

Condiciones de ejecución: El usuario debe estar autenticado.

Tabla 6. Caso de pruebas: Escenario Seleccionar Tipo de Contenido.

Escenario	Descripción	Manual de configuración	de	Respuesta del sistema	del	Flujo central

EC 3.1 Seleccionar Tipo de Contenido.	El usuario selecciona los tipo de contenido a exportar.	V	El sistema verifica que estén seleccionados los tipo de contenido, de ser así, se almacenarán sus datos en la base de datos.	El usuario selecciona la opción "Siguietes". El sistema muestra una interfaz para relacionar el tipo de contenido con los elementos del menú.
		Al menos un valor seleccionado.		
EC 3.2 Seleccionar Tipo de Contenido.	El usuario no selecciona ningún tipo de contenido a exportar.	I	El sistema verifica que estén seleccionados los tipo de contenido, de no ser así, el sistema muestra un mensaje indicando el error.	
		Ningún valor seleccionado.		

Tabla 7. Caso de pruebas: Escenario Seleccionar Elementos del Menú

Escenario	Descripción	Manual de configuración	Respuesta del sistema	Flujo central
EC 3.1 Seleccionar Elementos del Menú.	El usuario selecciona los elementos del menú a exportar.	V	El sistema verifica que estén seleccionados los elementos del menú, de ser así, se almacenarán sus datos en la base de datos.	El usuario selecciona la opción "Siguietes". El sistema muestra una interfaz para relacionar el tipo de contenido con los elementos del menú.
		Al menos un valor seleccionado.		
		I		

EC 3.2	El usuario no selecciona ningún elemento del menú a exportar.	Ningún valor seleccionado.	El sistema verifica que estén seleccionados los elementos del menú, de no ser así, el sistema muestra un mensaje indicando el error.	
--------	---	----------------------------	--	--

Tabla 8. Caso de Prueba: Escenario Enlazar Tipo de Contenido con Menú.

Escenario	Descripción	Manual de configuración	Respuesta del sistema	Flujo central
EC 3.1	El usuario enlaza los tipo de contenido con los menús seleccionados .	V Todos los tipo de contenido enlazados a un menú.	El sistema verifica que estén relacionados los elementos del menú y tipo de contenido, de ser así, se almacenará sus datos en la base de datos y se exportaran a SQLite, se copian las imágenes y archivos a la carpeta a exportar.	El usuario selecciona la opción "Siguietes". El sistema muestra una notificación según el suceso ejecutado.
EC 3.2	El usuario no enlaza todos los tipo de contenido con los menús	I Falta al menos un tipo de contenido por enlazar a un menú.	El sistema verifica que estén relacionados los elementos del menú y tipo de contenido de	

	seleccionados		no ser así, el sistema muestra un mensaje indicando el error.	
--	---------------	--	---	--

Resultados de las pruebas funcionales

Se probaron todas las funcionalidades que responden a los casos de pruebas del módulo y se obtuvieron un total de 18 no conformidades las cuales fueron todas resueltas, siendo los principales errores detectados en errores ortográficos, funcionalidad y textos en idioma inglés.

La Figura 7 desglosa las no conformidades detectadas en las tres iteraciones realizadas. En una primera iteración se recoge un total de 15 no conformidades de las cuales se resolvieron 8 y las restantes 7 se quedaron pendientes para la segunda iteración la cual arrojó un total de 8 no conformidades, resolviéndose un total de 10 quedando 5 pendiente, para la tercera iteración se detectaron 0 nuevas no conformidades y se solucionaron 5 teniendo en cuenta la que quedaba pendiente.

Dentro de las no conformidades encontradas dentro del módulo se tiene que la base de datos genera se duplicaba ya que al generar una nueva base de datos no se tuvo en cuenta borrar o suscribir la anterior. La misma fue resuelta borrando la base de datos antes de generar una nueva evitando guardar datos innecesarios. También se detectó la necesidad de brindar la posibilidad al usuario de regresar a la vista anterior en el proceso de selección de los datos a exportar. Para brindarle solución a la misma se diseñaron en cada vista los botones de “Atrás” y “Cancelar”. El primero le permite regresar a la vista anterior al usuario y elimina los datos de la base de datos para que pueda hacer una nueva selección. El botón “Cancelar” borra todos los datos presentes en las tablas creadas por el módulo de la base de datos y regresa al usuario a la página principal del sitio.

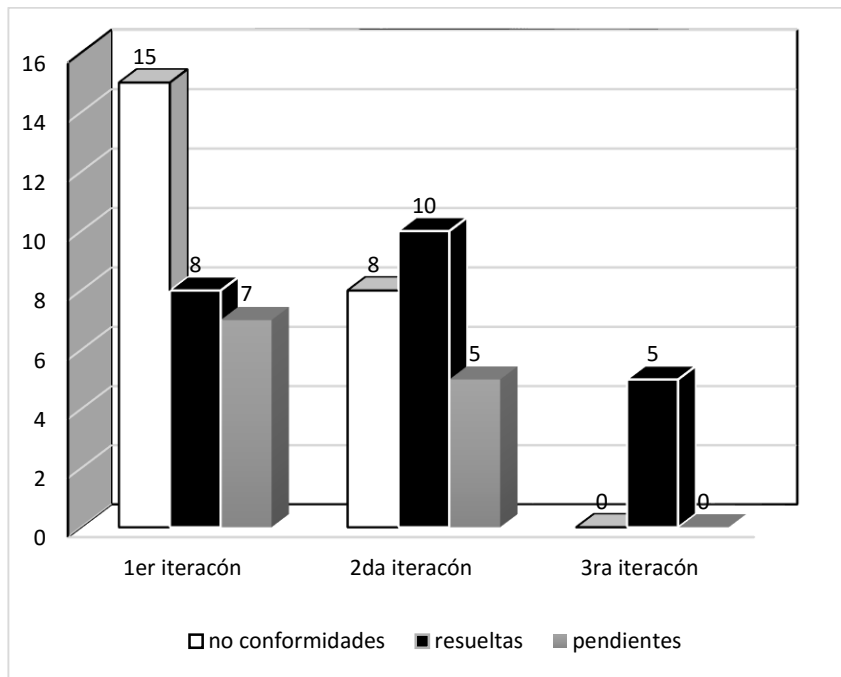


Figura 8. Resultados de las pruebas funcionales.

3.3.2 Seguridad

Resultados de las pruebas de seguridad

Las pruebas de seguridad son usadas para mitigar riesgos relacionados con la existencia de vulnerabilidades en aplicaciones web y la probabilidad de sufrir ataques de Hackers y/o usuarios malintencionados, que podrían descubrir y explotar esas vulnerabilidades causando daño al negocio (Salazar, 2012).

Las pruebas arrojaron que existían problemas en la validación de los datos dentro del sistema. De se han solucionado un 20% quedando pendiente el 10% de los mismos. También los datos demostraron que no existe problema en la autenticación del sistema.

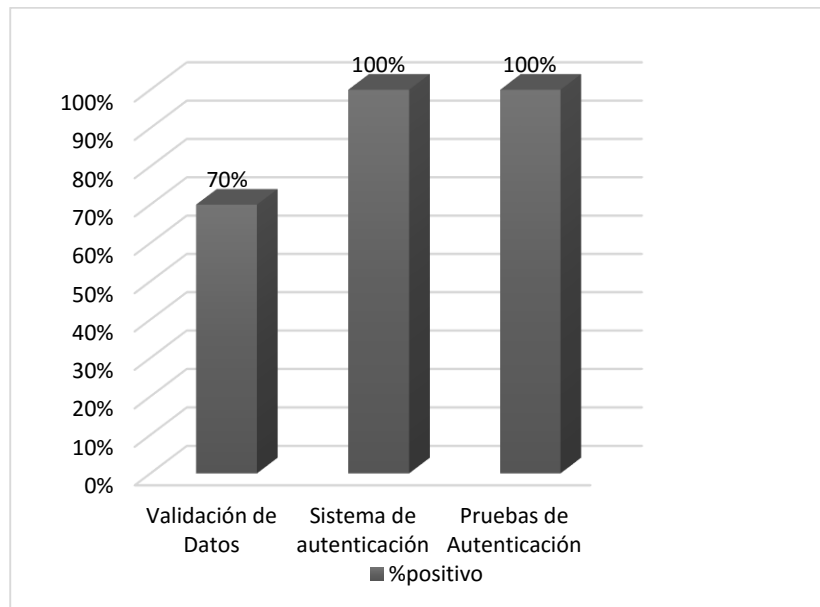


Figura 9. Resultados de las pruebas de seguridad.

3.3.3 Carga y Estrés

Prueba de carga: Las pruebas de carga miden la capacidad del sistema para continuar funcionando apropiadamente bajo diferentes condiciones de carga. Su meta es determinar y asegurar que el sistema funciona apropiadamente aún más allá de la carga de trabajo máxima esperada (HERNAN, 2005). En otros términos, esta prueba determina y valida la respuesta de la aplicación cuando es sometida a una carga de usuarios y/o transacciones que se espera en el ambiente donde se desplegará el sistema.

Prueba de estrés: Enfocada a evaluar cómo el sistema responde bajo condiciones extremas. (Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible) (HERNAN, 2005). El objetivo de este tipo de pruebas es conocer el límite de carga de trabajo al que puede llegar el sistema, lo que hace posible que se puedan tomar medidas antes de llegar a ese límite.

Resultados de las pruebas de carga y estrés.

Durante la realización de la prueba se desea simular el entorno donde debe interactuar el módulo, tanto del hardware como software, para obtener una correcta información de comportamiento y resultados en general. Por lo que se eligió un escenario con las siguientes características.

Hardware:

- Tipo de procesador: corei3, con una velocidad del procesador de 2.00 GHz
- Memoria: 4GB RAM.
- Tipo de Red: Ethernet 10/100Mbps.

Software:

- Tipo de Servidor Web: Apache v2.4.12.
- Máximo de hilos concurrentes (simulación de usuarios): 20
- Plataforma: Sistema operativo Ubuntu 12.4.1.
- Servidor de BD: MySQL 5.6.24.

Se añadirá el elemento “Informe Agregado” que brindará la siguiente información:

Etiqueta: El nombre de la muestra (conjunto de muestras).

Muestras: El número de muestras para cada URL (peticiones).

Mín: El mínimo tiempo transcurrido para las muestras de la URL dada.

Máx: El máximo tiempo transcurrido para las muestras de la URL dada.

% Error: Porcentaje de las peticiones con errores.

Rendimiento: Rendimiento medido en base a peticiones respondidas por segundo/minuto/hora.

Usuarios: Cantidad de usuarios concurrentes.

Tabla 8. Prueba de carga y estrés.

# Muestras	Mín	Máx	% Error	Rendimiento (peticiones/segundos)	Usuarios
400	2 segundos	40 segundos	0	50	10
550	2 segundos	46 segundos	0	46	15
680	3 segundos	52 segundos	0	44	20

Resultados de la Prueba de Carga y Estrés.

El módulo solo podrá ser utilizado por usuarios administradores de los sitios web o por un susuario administrador con rol específico de editor, por ello se determina una concurrencia baja de hasta 20 usuarios. Después de las simulaciones realizadas, se obtiene 0 % de error de las peticiones realizadas al sistema con un rendimiento de variable según la cantidad de usuarios concurrentes pero que se encuentran dentro del rango definido en los requisitos no funcionales ya que procesa hasta 10 peticiones por segundo y responde en menos de un minuto en cada caso, quedando una prueba satisfactoria.

3.4 Conclusiones parciales

Como resultado de este capítulo se obtuvo el diagrama de componentes el cual muestra la estructura física que tiene el componente de software. Se logró de esta manera obtener el código del módulo Descarga apk y de la aplicación WebOffline que consumiera la base de datos generada por el módulo, basándose en el análisis y diseño de los capítulos anteriores. Además, para validar la completitud de respuesta a los requerimientos identificados se realizaron pruebas de caja negra al módulo además de pruebas de carga y estrés. El resultado de estas pruebas arrojó que el mayor número de dificultades se encontraba en la parte ortográfica y además demostró que el sistema es capaz de trabajar sobre diferentes sitios desarrollados sobre Drupal 7, sin importar el gestor de base de datos que utilicen. También las pruebas de carga y estrés demostraron que el sistema se encuentra en condiciones de ser desplegado sobre el entorno para el cual fue desarrollado.

Conclusiones Generales

Una vez completada la presente investigación, se puede concluir que:

- Con el estudio del estado del arte se identificaron una serie de características deseables de forma general, pero se arribó a la conclusión de que no existe una herramienta que satisfaga por completo las necesidades planteadas y por ello se proyectó una solución informática nueva para resolver el problema de investigación. Este estudio también permitió ir delineando un conjunto posible de herramientas y tecnologías para la implementación de la solución.
- Se modelaron los artefactos que contribuyeron al diseño de la propuesta de solución, posibilitando un mayor soporte a la implementación de los requisitos identificados; logrando una mayor organización estructura del código fuente, disminuyendo el impacto ante futuras modificaciones.
- El diseño y aplicación sistemas de pruebas de software permitió erradicar insuficiencias presentes en la solución, con vista a lograr un producto más seguro, funcional y de mayor calidad.
- Se implementó una solución informática compuesta por un módulo que permite el empaquetado del contenido presente en sitios web desarrollados en Drupal 7, en conjunto a una aplicación Android que lee y muestra el contenido de estos paquetes para asegurar la contribuir a la divulgación de esta información a través de tecnologías móviles.

Recomendaciones

1. Desarrollar funcionalidades que optimicen los tiempos de respuesta para el exportado al lenguaje SQLite.
2. Implementar las funcionalidades necesarias para permitir a la aplicación conectarse a un portal web desarrollado sobre Drupal 7 y descargar o actualizar la base de datos necesitada.

Referencias

Acunetix. 2016. acunetix. [En línea] Acunetix, 2016. [Citado el: 2016 de febrero de 2016.]

Álvarez, Miguel Ángel. 2011. DesarrolloWeb. [En línea] 2011. [Citado el: 13 de 11 de 2015.] <http://DesarrolloWeb.com>.

Buytaert, Dries. 2016. Drupal. *Coding standards*. [En línea] 8 de enero de 2016. [Citado el: 04 de abril de 2016.] <https://www.drupal.org/coding-standards>.

CMS Usage Statistics. 2013. built whit. *Content Management System Distribution*. [En línea] 2013. [Citado el: 17 de noviembre de 2015.] <http://trends.builtwith.com/cms>.

ComponentSource. 2014. ComponentSource. [En línea] 2014. [Citado el: 29 de 9 de 2014.] <http://www.componentsource.com/services/about-us/components-es.html>.

Cubaperiodistas, Redacción de. 2010. Cubaperiodistas. *Cubaperiodistas*. [En línea] Unión de Periodistas de Cuba, 2010. [Citado el: 1 de 03 de 2016.] <http://www.cubaperiodistas.cu/index.php/2016/03/cuba-por-gobernanza-de-internet-democratica-y-participativa/>.

Drupal. 2013. Características de Drupal. *Drupal*. [En línea] 2013. [Citado el: 17 de noviembre de 2015.] <http://drupal.org.es/caracteristicas>.

EcuRed. 2011. EcuRed. *EcuRed*. [En línea] EcuRed, 2011. [Citado el: 28 de marzo de 2016.] http://www.ecured.cu/Eclipse,_entorno_de_desarrollo_integrado.

Erich Gama, Richard Helm, Ralph Johnson, Jhon Vlissides. 2003. *Patrones de diseño. Elementos de software orientado a objetos reutilizable*. Madrid : Pearson Educación, 2003.

España, Ministerio de Educacionde. Navegación offline. *Instituto de Tecnología Educativa*. [En línea] [Citado el: 30 de octubre de 2015.] http://www.ite.educacion.es/formacion/materiales/157/cd/m2_1_navegacion/navegacin_offline.html.

ETECSA. 2015. Abrirán salas de wi-fi en 35 puntos del territorio cubano, anuncia ETECSA. *Cubadebate*. [En línea] 18 de junio de 2015. [Citado el: 15 de septiembre de 2015.] <http://www.cubadebate.cu/noticias/2015/06/18/abriran-salas-de-wi-fi-en-35-puntos-del-territorio-cubano-anuncia-etecsa/>.

—. 2012. ETECSA. *Empresa de Telecomunicaciones de Cuba S.A.* [En línea] 2012. [Citado el: 1 de 03 de 2016.] http://www.etcscsa.cu/?page=internet_conectividad&sub=internet.

Falcón, José Antonio Caballar. 2010. *Wi-Fi.Lo que necesita conocer*. Madrid : CR Libros, 2010.

Flores, Anny. 2012. *Desarrollo de Sistema Transaccional Para Manejo de Reportes del Proceso de Inyección de Químicos en la Superintendencia de tratamiento y calidad de fluidos PDVSA distrito Morichal*. Panama : Edo Monagas, 2012.

Foundation, Apache Software. 2011. Apache. *Apache JMeter*. [En línea] 09 de mayo de 2011. [Citado el: abril de 05 de 2016.] <http://jmeter.apache.org/>.

Gilfillian, Ian. 2010. *La biblia del MySQL*. 2010.

Guevara, Jorge Martínez Ladrón de. 2012. *Fundamentos de programación en Java*. Madrid : EME, 2012.

Inc Nginx. 2011. NGINX. *nginx.org*. [En línea] 2011. [Citado el: 10 de noviembre de 2015.] <https://www.nginx.com/>.

Ivar G. B. Jacobson, James Rumbaugh. 2000. *Proceso Unificado de Desarrollo De Software*. Madrid : s.n., 2000.

Joomla. 2005. Open Source Matters. [En línea] 2005. [Citado el: 17 de noviembre de 2015.] <http://www.joomla.org/core-features.html>.

Kauler, Leto. 2015. HTTrack website copier. [En línea] 2015. [Citado el: 1 de diciembre de 2015.] <https://www.httrack.com>.

Knowledge management: where did it come from and where will it go? **Wiig, Karl M. 1997.** 1, s.l. : Expert systems with applications, 1997, Vol. 13, págs. 15-27.

Knutson, Randall. 2015. Drupal. [En línea] 11 de junio de 2015. [Citado el: 1 de diciembre de 2015.] <https://www.drupal.org/search/site/%E2%80%A2%09Static%20Generator>.

Lapuente, María Jesús Lamarca. 2013. Hipertexto, el nuevo concepto de documento en la cultura de la imagen. *hipertexto.info*. [En línea] 8 de diciembre de 2013. [Citado el: 15 de Septiembre de 2015.] <http://www.hipertexto.info/documentos/web.htm>.

Larman, Craig. 2003. *UML y Patronos*. 2003. 2a Edición.

- Letelier, Patricio. 2005.** *Metodologías Ágiles y XP*. Valencia : Departamento de Sistemas Informáticos y Computación, 2005.
- Luca, Juan Ignacio de Tena. 2004.** *Manual Imprescindible de PHP5*. Madrid : GRUPO ANAYA, S.A., 2004.
- MDA vs Factorías de Software.* **Muñoz, Javier y Pelechano, Vicente. 2005.** 1, s.l. : Actas del II Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones (DSDM 2005). Granada (Spain), 2005, Vol. 13.
- Miani, Antonio Aliaga y Marco. 2011.** I.E.Sanvicente. *PosgreSQL*. [En línea] 2011. [Citado el: 15 de noviembre de 2015.] <http://www.iessanvicente.com/colaboraciones/postgreSQL.pdf>.
- Montiel, Daniel Ponsoda. 2008.** *Introducción a SQLite*. s.l. : Creative Commons , 2008.
- Naramore, Elizabeth. 2005.** *Beginning PHP5, Apache, and MySQL Web Development*. Canadá : Wiley Publishing, 2005.
- NetBeans Community. 2000.** NetBeans Community. [En línea] 2000. [Citado el: 11 de 11 de 2015.] <http://netbeans.org>.
- Oliva, Manuel Castells. 2009.** *COMUNICACIÓN Y PODER*. Madrid : Alianza Editorial,, 2009.
- Ollendyke, Bryan. 2014.** Drupal. [En línea] 2 de diciembre de 2014. [Citado el: 1 de diciembre de 2015.] <https://www.drupal.org/search/site/HTML%20Export>.
- OMG. 2015.** OMG. *Introduction to OMG's* . [En línea] 24 de junio de 2015. [Citado el: 11 de 11 de 2015.] http://www.omg.org/gettingstarted/what_is_uml.htm.
- Orozco, David. 2011.** Definición de Android. *ConceptoDefinicion.de*. [En línea] 1 de junio de 2011. [Citado el: 30 de octubre de 2015.] <http://conceptodefinicion.de/android/>.
- PgAdmin III. 2008.** PgAdmin III. [En línea] 10 de marzo de 2008. [Citado el: 17 de noviembre de 2015.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.
- Popkin Software and Systems. 2007.** tldp. [En línea] 19 de 6 de 2007. [Citado el: 11 de 11 de 2015.] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
- Pressman, Roger. 2003.** *Ingeniería de Software, un enfoque práctico*. New York : s.n., 2003.
- Redacción de Cubaperiodistas.cu. 2010.** Cubaperiodistas. *Cuba por gobernanza de Internet democrática y participativa*. [En línea] Unión de Periodistas de Cuba., 2010. [Citado el: 01 de 03 de 2016.]

<http://www.cubaperiodistas.cu/index.php/2016/03/cuba-por-gobernanza-de-internet-democratica-y-participativa/>.

Rios, Deysi Yuri Hernández. 2009. Problemas comunes en el desarrollo y uso de aplicaciones distribuidas. [En línea] 2009. [Citado el: 15 de noviembre de 2015.] <http://dysihdez.blogspot.com>.

Robertson, James. 2008. Step Two Designs Pty Ltd. [En línea] 2008. [Citado el: 15 de noviembre de 2015.] http://www.steptwo.com.au/papers/kmc_what.

Salazar, Edgar D. 2012. OWASP. *Pruebas de Seguridad en aplicaciones web segun OWASP*. [En línea] 2012. [Citado el: 2016 de abril de 05.] https://www.owasp.org/images/2/2f/OWASP_SUSCERTE.pdf.

Schlachter, Alexis. 2008. Internet en Cuba : año diez. *Granma*. 2008.

SeeD. 2015. SeeD. *Arquitectura de Drupal 7*. [En línea] SeeD EM, 2015. [Citado el: 7 de 1 de 2016.] <http://www.seedem.co/es/blog/arquitectura-de-drupal-7>.

The Eclipse Foundation. 2016. Ecilpse. *About the Eclipse Foundation*. [En línea] 2016. [Citado el: 2016 de abril de 04.] <https://eclipse.org/org/>.

Turner, Raymond. 2005. *The Foundations of Specification*. s.l. : Journal of Logic and Computation, 2005. Vol. 15.

Tusa, Stalin. 2012. slideshare. *Ingeniería de software I Metodología OpenUp*. [En línea] LinkedIn Corporation, 2012. [Citado el: 17 de noviembre de 2015.] <http://www.slideshare.net/StalinTuza/ingenieriasoftwareopenup#btnPrevious>.

UIT. 2015. Unión Internacional de Telecomunicaciones. *Measuring the Information Society Report*. [En línea] 2015. [Citado el: 01 de 03 de 2016.] https://www.itu.int/en/ITU-D/Statistics/Documents/publications/mis2014/MIS2014_without_Annex_4.pdf.

Valdés, Damián Pérez. 2007. Maestros del Web. *¿Qué es Javascript?* [En línea] 3 de julio de 2007. [Citado el: 10 de noviembre de 2015.] <http://www.maestrosdelweb.com/que-es-javascript>.

Villena, Antonio Moreno y. 2009. AEEMT. [En línea] 2009. [Citado el: 10 de noviembre de 2015.] http://www.aeemt.com/contenidos_socios/Informatica/Informac_Informat_Tecnolog/AMV_AGI_AEEMT_HTML_Historia.pdf.

Visual Paradigm. 2014. Visual Paradigm. *Visual Paradigm*. [En línea] Paradigm, 2014. [Citado el: 5 de febrero de 2016.] <https://www.visual-paradigm.com/>.

W3Schools. 2011. W3Schools.com. *CSS Tutorial*. [En línea] 2011. [Citado el: 10 de noviembre de 2015.] <http://www.ecured.cu/index.php/CSS>.

WordPress. 2013. Características de WordPress. [En línea] 2013. [Citado el: 17 de noviembre de 2015.] http://codex.wordpress.org/es:Caracter%C3%ADsticas_de_WordPress.

Bibliografía Consultada

Autores. 2012. *Gran libro de Android.* 2012.

Alegsa, Leandro. 2016. ALEGSA.com.ra. *ALEGSA.com.ra.* [En línea] ALEGSA, 2016. [Citado el: 5 de enero de 2016.] <http://www.alegsa.com.ar/Dic/sghd.php>.

Centro de Apoyo Tecnológico a Emprendedores. 2012. *Estudio de los sistemas de gestión de contenido.* Castilla : Creative Commons By - Sa, 2012.

Formantín, Yusleidys Pérez y Sierra, Indira Peraza. 2007. *GUÍA PARA EL TRABAJO CON EL CMS DRUPAL.* Habana : s.n., 2007.

Kane, Victor. 2009. *Leveraging Drupal: Getting Your Site Done Right.* Canada : Wiley Publishing, 2009.

Soriano, Jose Enrique Amaro. 2014. *El gran libro de programación avanzada con Android.* s.l. : Alfaomega, 2014.

Sparx Systems. 2007. Sparx Systems. *Diagrama de Despliegue UML 2.* [En línea] 2007. [Citado el: 10 de diciembre de 2015.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

Vandyke, Jhon, Westgate, Matt y Tomlinson, Todd. 2010. *Pro Drupal 7 Development.* s.l. : Apress, 2010.

Vásquez, Eveling Giselles Cruz. 2013. Slideshare. *Slideshare.* [En línea] LinkedIn Corporation, 30 de noviembre de 2013. [Citado el: 3 de diciembre de 2015.] <http://es.slideshare.net/EvelingGiselleCruzVs/metodologia-monografia>.

Valls, Martínez. 2012 *Android en la era digital alcance de tus dispositivos favoritos.* 46870, Valencia : Ontinyent, Vol. 56.

Glosario de términos

WWW: Red informática mundial, sistema lógico de acceso y búsqueda de la información disponible en Internet, cuyas unidades informativas son las páginas web.

Wi-Fi: Tecnologías de comunicación inalámbrica mediante ondas, también llamada WLAN o estándar IEEE 802.11. No es una abreviatura de Wireless Fidelity, simplemente es un nombre comercial (Falcón, 2010).

Offline: La navegación sin conexión, permite visitar un sitio web sin conexión a Internet teniendo descargado todo el contenido del mismo.

Zapia: Aplicación que permite compartir imágenes, videos, documentos y otras aplicaciones entre distintas terminales.

PHP: Se trata de un lenguaje de scripting para la programación de páginas dinámicas de servidor. Es un lenguaje de tipo gratuito, y forma parte del software que se conoce como de código abierto (Open Source).

Hooks (ganchos): Se utilizan en Drupal como unas funciones que permiten la comunicación entre los distintos módulos del sistema.

JavaScript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

CCS: Es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora o por voz.

Memcached: Es un sistema distribuido de propósito general para caché basado en memoria, diseñado por Danga Interactive y que es muy usado en la actualidad por múltiples sitios web.

Canvas: Es un elemento de HTML 5 que permite dibujar gráficos, manipular imágenes y realizar animaciones en una página web de forma dinámica. Es decir, el gráfico se crea en el momento en que se carga la página.

Pligins: Un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

APIs: Application Program Interface es conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.

GNU: Licencia Pública General de GNU o más conocida por su nombre en inglés *GNU General Public License* (o simplemente sus siglas del inglés GNU GPL) fue un proyecto iniciado por Richard Stallman con el objetivo de crear un Sistema operativo libre.

OMG: Es un consorcio, formado en 1989, dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA y BPMN.

GPL: Es la licencia más ampliamente usada¹ en el mundo del software y garantiza a los usuarios finales la libertad de usar, estudiar, compartir (copiar) y modificar el software.

GUI: La interfaz gráfica de usuario, conocida también como GUI (del inglés *graphica luser interface*) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el Sistema operativo de una máquina o computador.

Multihilo: Se trata en la cual los procesos tienen diversos hilos de ejecución. Si uno de ellos deja de correr no se cae toda la aplicación, sino que simplemente se corta esa vertiente. Se aplica a procesos susceptibles de repartir su trabajo entre los distintos núcleos y/o procesadores disponibles en el sistema.

Rollbacks: Son importantes para la integridad de la base de datos, a causa de que significan que la base de datos puede ser restaurada a una copia limpia incluso después de que se han realizado operaciones erróneas.

Empotrados: Es un sistema electrónico diseñado para realizar pocas funciones en tiempo real, según sea el caso. Se diseñan para cubrir necesidades específicas.

BSD: Distribución de Software Berkeley, es una licencia de software libre.

MVCC: Control de la Concurrencia Multi Versión modo de administrar versiones de archivos que permite trabajar fácilmente a más de un desarrollador en el mismo proyecto.

Proxy inverso: Es un servidor proxy que funciona en representación de un servidor. Un servidor proxy inverso toma de peticiones de los navegadores web en todo el Internet, dichas solicitudes generan una respuesta de vuelta al servidor proxy inverso, que luego se remitirá al navegador web.

Servidor proxy: Es un equipo que actúa de intermediario entre un explorador web (como Internet Explorer) e Internet. Los servidores proxy ayudan a mejorar el rendimiento en Internet ya que almacenan una copia de las páginas web más utilizadas.

Anexos

Anexo N° 1. Entrevista

Tabla 9. Entrevista aplicada a directivos del centro CIDI

Numero	Pregunta
1	¿Qué tecnologías se utilizan para el desarrollo de portales web en el centro?
2	¿Cómo se compone la estructura de almacenamiento de Drupal?
3	¿Qué es un Tipo de Contenido y como está compuesto?
4	¿Qué tecnología utiliza Drupal? ¿Por qué?
5	¿Cómo se implementa la seguridad de los portales desarrollados en Drupal?

Anexo N° 2. Diagramas de Clases del Diseño

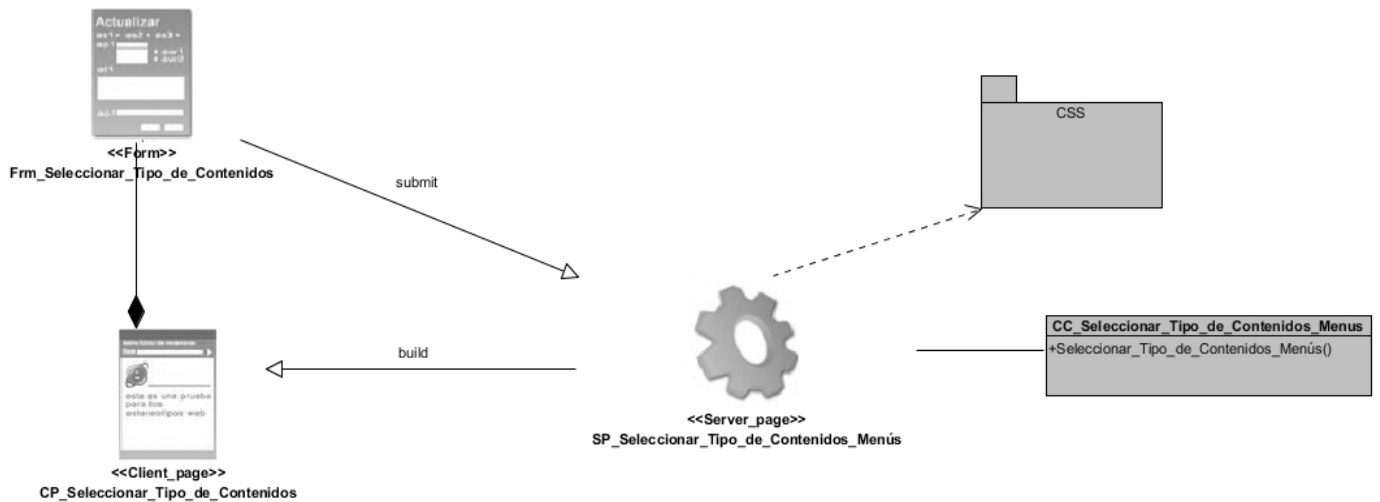


Figura 10. Diagrama de clases de diseño. Seleccionar Tipo de Contenido y Seleccionar Menús.

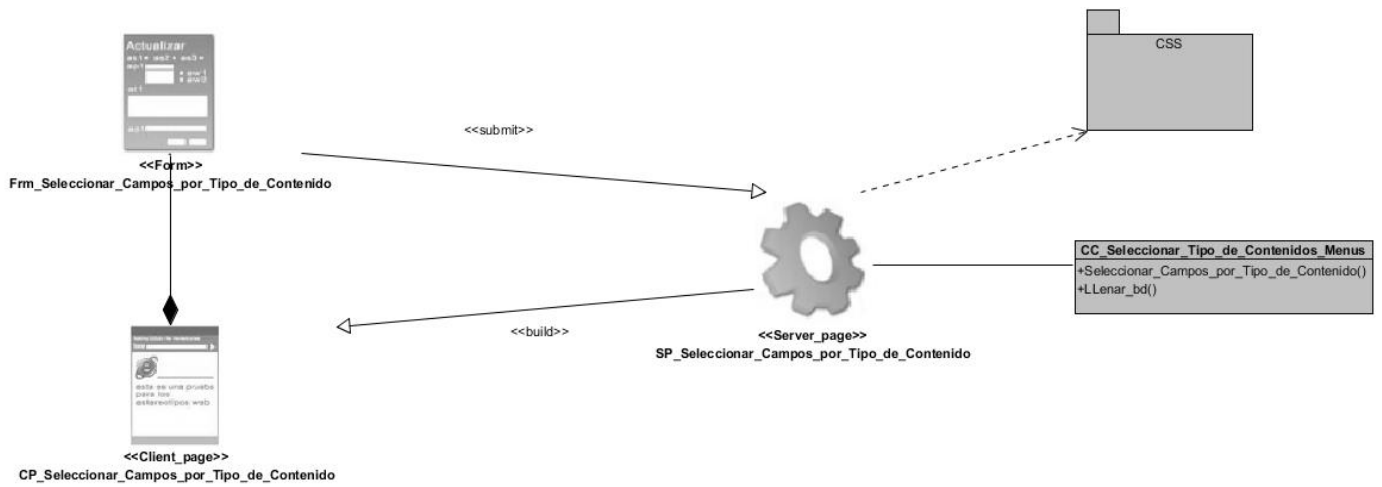


Figura 11. Diagrama de clases de diseño. Seleccionar Campos por Tipo de Contenido.

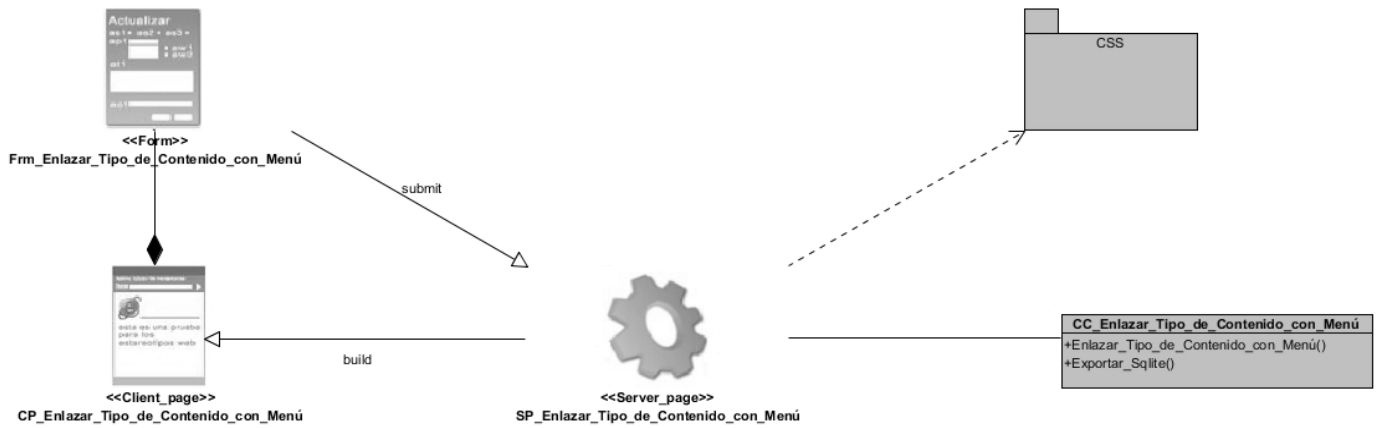


Figura 12. Diagrama de clases de diseño. Enlazar Tipo de Contenido con Menú.



Figura 13. Diagrama de clases de diseño. Aplicación.

Anexo N°. 3. Pantallas de la Aplicación

Módulo Descarga_apk:

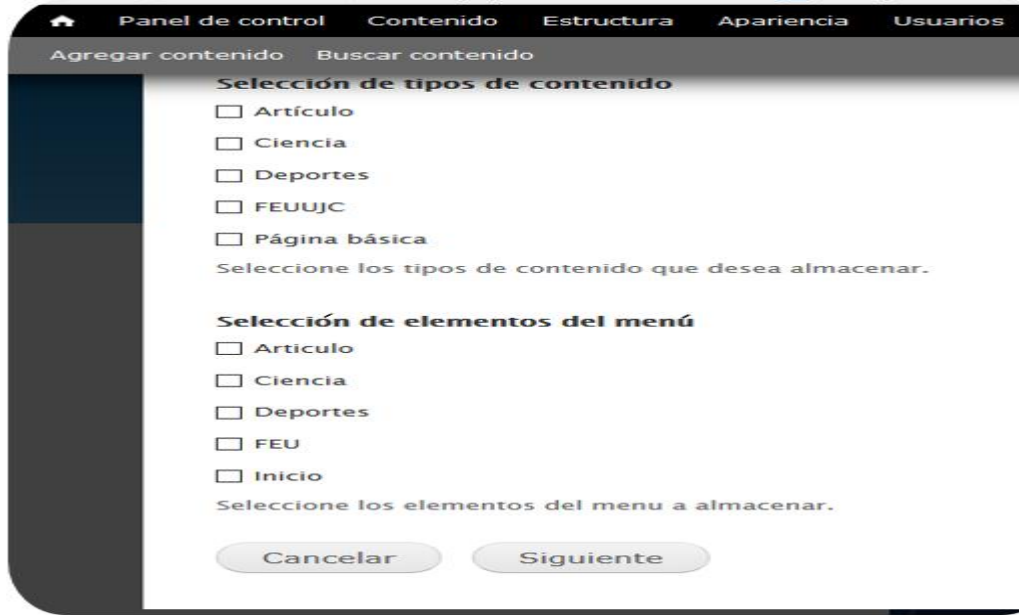


Figura 14. Seleccionar Tipos de Contenidos y Menús.

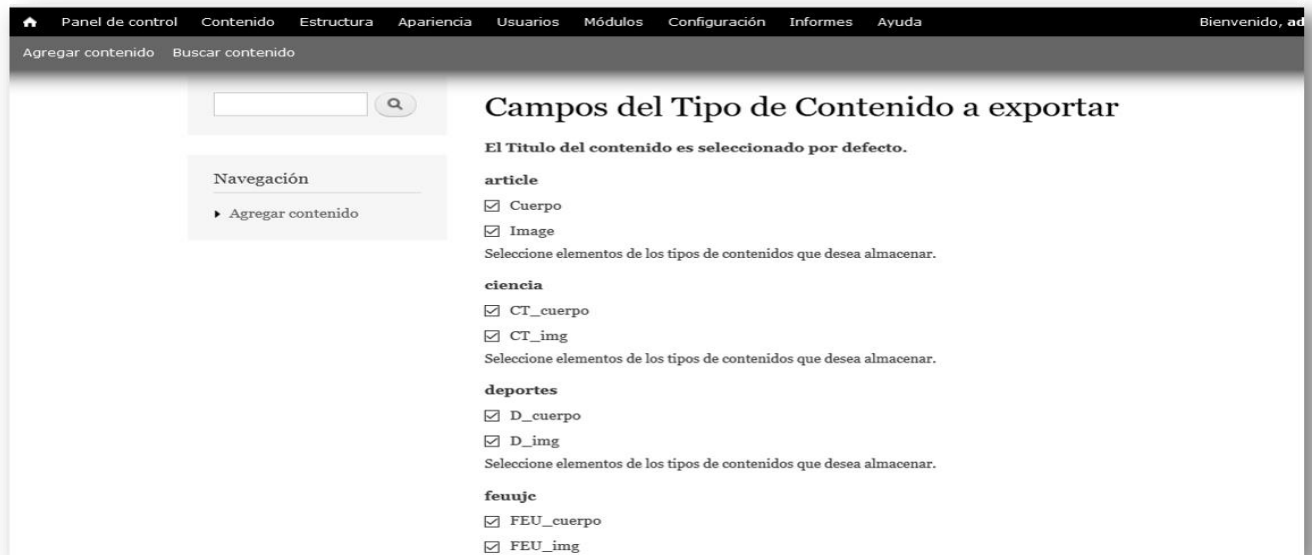


Figura 15. Campo del Tipo de Contenido a exportar.

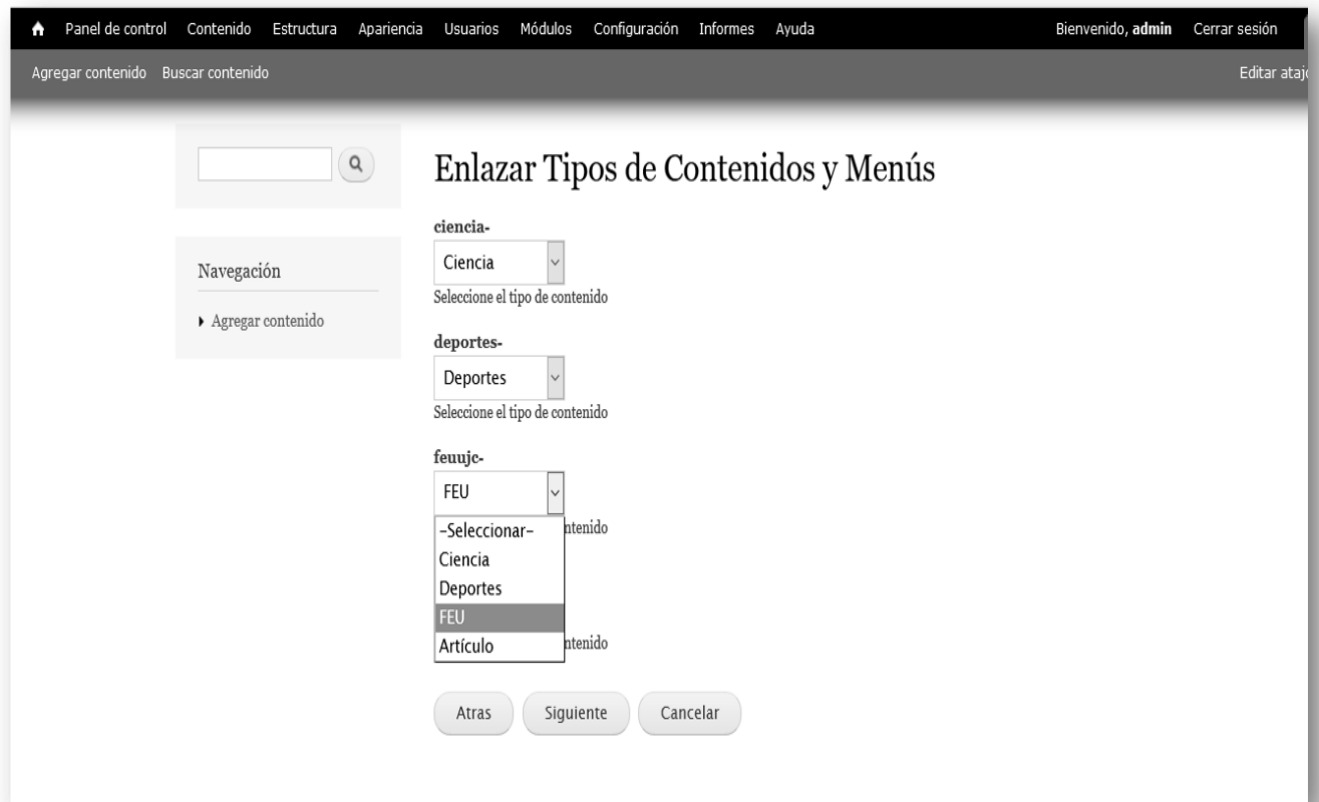


Figura 16. Enlazar Tipos de Contenidos y Menús.