

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1



Sistema para el empaquetado web de contenido en Wordpress

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO
DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autor: Rolando Vázquez Conyedo

Tutores:

Ing. Yuriesky Madrigal Vilches

Lic. Katuska Cedeño Espinosa

La Habana, 8 de June de 2017

DECLARACIÓN DE AUTORÍA

Declaro ser 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 _____.

Rolando Vázquez Conyedo

Firma del Autor

Ing. Yuriesky Madrigal Vilches

Lic. Katuska Cedeño Espinosa

Firma del Tutor

Firma del Tutor



“El único modo de hacer un gran trabajo es amar lo que haces. Si no lo has encontrado todavía, sigue buscando. No te acomodes. Como con todo lo que es propio del corazón, lo sabrás cuando lo encuentres”

Steve Jobs

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.

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 mi novia y mis amigos que han sido un soporte en el cual apoyarme cuando me ha hecho falta. 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.

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, propone el desarrollo de un plugin que permita obtener la información presente en sitios desarrollados sobre *Wordpress* 4.7.2 y a su vez permita el empaquetado de la información en forma de base de datos 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; lo que posibilitará que un mayor número de usuarios pueda tener acceso a la información de estos sitios de manera *offline*. 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 lenguaje de programación y como entornos de desarrollo integrados NetBeans y AndroidEstudio.

Palabras clave: Android; empaquetado; *offline*; *plugin*; tecnologías móviles; *Wordpress*.

Índice de Contenido

INTRODUCCIÓN	1
CAPÍTULO 1: Fundamentación teórica.....	7
1.1. Conceptos asociados al dominio del problema	7
1.2.1. Internacionales	9
1.2.2. Nacionales.....	10
1.3. Análisis de las tecnologías, herramientas y lenguajes a utilizar	11
1.3.1. Metodología de desarrollo de <i>software</i>	11
1.3.2. Lenguajes utilizados	14
1.3.3. Entornos de Desarrollo	16
1.3.4. Sistemas gestores de base de datos	17
1.3.5. Servidor web.....	19
1.3.6. Herramientas	20
1.4. Conclusiones parciales	22
CAPÍTULO 2: Análisis y diseño del sistema.	24
2.1. Introducción	24
2.2. Modelo de dominio	24
2.2.1. Descripción de las clases del Modelo de Dominio	25
2.3. Levantamiento de requisitos	26
2.3.1. Requisitos Funcionales y Requisitos no Funcionales.....	26
2.3.2. Historias de Usuario	29
2.4. Arquitectura	31
2.4.1. Patrones de Diseño	33
2.5. Diseño	35

2.5.1. Diagrama de Clases de Diseño	35
2.6. Modelo de datos	38
2.7. Modelo de despliegue	38
2.8. Conclusiones parciales.....	39
3.1. Diagrama de componentes.....	40
3.2. Estándares de codificación.....	43
3.3. Pruebas.....	44
3.3.1. Funcionales (Para el plugin y la aplicación android)	44
3.3.2. Carga y Estrés (para el plugin)	47
3.3.3. Unitarias (para la aplicación android).....	49
3.4. Conclusiones parciales.....	50
CONCLUSIONES.....	52
RECOMENDACIONES	53
REFERENCIAS.....	54
GLOSARIO DE TÉRMINOS	64
ANEXOS	68
Anexo N° 1. Entrevista.....	68
Anexo No 2. Diagramas de Clases del Diseño.....	68
Anexo N°. 3. Pantallas del plugin.....	69



Índice de Tablas

<i>Tabla 1. Requisitos Funcionales.</i>	<i>27</i>
<i>Tabla 2. Descripción de HU: Mostrar contenido.</i>	<i>30</i>
<i>Tabla 3. Estándar de codificación del Wordpress.....</i>	<i>43</i>
<i>Tabla 4. Caso de pruebas: Escenario Mostrar Contenido.</i>	<i>45</i>
<i>Tabla 5. Entrevista aplicada a directivos del centro CIDI.....</i>	<i>68</i>

Índice de Figuras

<i>Figura 1. Diagrama de Clases del Modelo de Dominio.....</i>	<i>25</i>
<i>Figura 2. Arquitectura de Wordpress (Morante Arreaga, 2015).</i>	<i>32</i>
<i>Figura 3. Diagrama de Clases de Diseño de Wordpress.....</i>	<i>36</i>
<i>Figura 4. Diagrama de clases de diseño. Seleccionar Campos por Tipos de Contenido.</i>	<i>37</i>
<i>Figura 5. Modelo relacional.</i>	<i>38</i>
<i>Figura 6. Diagrama de Despliegue.....</i>	<i>39</i>
<i>Figura 7. Diagrama de Componentes de la Aplicación.....</i>	<i>40</i>
<i>Figura 8. Resultados de las pruebas funcionales para el plugin.....</i>	<i>46</i>
<i>figura 9. Resultados de las pruebas funcionales para el plugin.....</i>	<i>47</i>
<i>Figura 10. Diagrama de clases de diseño. Seleccionar Tipo de Contenido.....</i>	<i>68</i>
<i>Figura 11. Diagrama de clases de diseño. Seleccionar Campos por Tipos de Contenido.....</i>	<i>69</i>
<i>Figura 12. Modificar título de contenido a descargar.....</i>	<i>69</i>
<i>Figura 13. Seleccionar Campos del Tipo de Contenido a exportar.....</i>	<i>70</i>
<i>figura 14. Apliación Android Wordpress contigo.....</i>	<i>70</i>

INTRODUCCIÓN

En los últimos años el incremento de aplicaciones web en Internet ha sido notable. Según datos estadísticos difundidos por www.worldwidewebsize.com, en septiembre de 2013 Internet estaba compuesta por más de 8.700 millones de páginas web (Narom, 2013) y en enero 2017 por más de 199.54 millones (Narom, 2017); lo que demuestra que el cúmulo de información que se va generando es cada vez mayor.

Para la creación de aplicaciones web, los desarrolladores emplean diferentes Sistemas Gestores de Contenidos (CMS por sus siglas en inglés). Entre ellos están Drupal, Joomla, Wordpress y otros. Datos brindados por la página web w3techs.com, demuestran que el CMS más usado hasta 2017 es Wordpress con un 59.5 % (Fawkes, 2017), lo que expresa que más de la mitad de las páginas web de los últimos años han sido desarrolladas sobre este CMS.

Datos compartidos por la agencia *We Are Social* con un amplio estudio sobre redes sociales e Internet, demuestran que existían en el 2015 más de 7.210 billones de habitantes en el planeta, de ellos el 51 % utilizan dispositivos móviles, que representa unos 3.649 billones de personas y el 42 % tienen acceso a Internet, que equivalen a 3.010 billones de personas. Regularmente usan las redes sociales un 28 %, de ellos el 23 %, que es igual a 1.968 billones de personas, se conectan a ellas a través de sus dispositivos móviles. En el 2016 existían más de 7.395 billones de habitantes del planeta, de ellos el 51 % utilizan móviles (3.790 billones de personas) y el 46 %, unos 3.419 millones de personas, son usuarios de Internet. Se conoce que el 31 % de la población mundial interactúa con las redes sociales, de ellos el 27 % (1.968 billones de personas) se conectan a ellas a través de sus dispositivos móviles (Redes Sociales, 2016).

En comparación el 2016 con el 2015 muestra una evolución de un 10 % de los usuarios de Internet (332 millones de personas). El número de usuarios únicos con dispositivos móviles creció en un 4 %, o sea, 141 millones de usuarios más. El número de usuarios en medios sociales también aumentó en un 10 % (219 millones de personas); mientras que los usuarios que se conectan a través del móvil creció en un 17 % con 283 millones de nuevos usuarios (Redes Sociales, 2016).

Mediante datos estadísticos extraídos de Internet Live Stats, en el 2016 se tiene conocimiento de países con acceso a Internet por encima de la media mundial (48.2 %), donde más de la mitad de su población posee acceso a la red de redes. Entre estos países se encuentra Estados Unidos con un 88,5 %, España con un 82,2 %, China con un 52,2 %, entre otros (Moreno, 2017); mientras que por debajo de la media se encuentran países como Cuba, donde la posibilidad de navegar en Internet ha crecido en el 2016, a pesar de ser un país de poco acceso a las redes. En total ya son 1 006 puntos públicos distribuidos en toda la isla desde los que se puede acceder a Internet, cifra que no incluye a instituciones, centros educativos o científicos, entre otros ya conectados (Guevara, 2017).

En el 2015 se facilitaron por todo el país puntos de acceso a Internet a la población mediante la puesta en servicios de 65 zonas Wi-Fi. En el 2016 se ensamblaron 252 nuevas zonas para un total de 317. Hasta el presente se encuentra cubierto el 97.6 % de todos los municipios del país con al menos una de esas zonas (Vladia, 2017). Existen además 193 salas de navegación de la Empresa de Telecomunicaciones de Cuba (ETECSA) en toda la geografía cubana, así como otras 613 de terceros, las cuales pueden ubicarse en hoteles, aeropuertos, sedes de los Joven Club de Computación y Electrónica, sitios del Ministerio de Salud o de Correos de Cuba, entre otros (Guevara, 2017).

Según datos ofrecidos por ETECSA al cierre de noviembre de 2016 se encontraban activas 1,4 millones de cuentas permanentes, 1,2 millones de cuentas de correo electrónico que acceden desde el móvil, 327 cuentas de navegación nacional y 2,7 millones de cuentas temporales (ETECSA, 2017).

Estas alternativas han logrado gran aceptación por parte de la población, pero el acceso es aún limitado por la disponibilidad del servicio teniendo en cuenta la cantidad de usuarios presentes en la isla que acceden simultáneamente. También presenta como desventaja que los usuarios tienen que trasladarse hacia estos puntos para obtener sus servicios, no pudiendo acceder desde sus casas o centros laborales. Todo lo planteado anteriormente limita a los clientes de acceder a la información que precisan regularmente, por lo que la creación de aplicaciones *offline* ha sido una alternativa para la interacción de diversos usuarios y la divulgación y distribución de información. Algunas de esas aplicaciones son el Formulario Nacional de Medicamento, la Guía de ETECSA y Ruta Habana.

En la Universidad de las Ciencias Informáticas (UCI) existen diferentes facultades que tienen asociados centros de desarrollo de *software*. El Centro de Ideoinformática (CIDI) perteneciente a la Facultad 1, que posee entre sus tareas principales la creación de sitios web, ha identificado la necesidad de crear una aplicación Android, que permita distribuir información sin conexión, utilizando las tecnologías móviles existentes en el país que emplean este sistema operativo y que a su vez posibilite a un mayor número de usuarios acceder a la información presente en cualquier sitio web.

A la necesidad de crear este tipo de aplicación, se unen las problemáticas del acceso a Internet descritas anteriormente y que aún persisten en Cuba; se reconoce además a *Wordpress* como uno de los CMS más usados a nivel mundial, el aumento del uso de las tecnologías móviles con Sistema Operativo Android y la existencia de sectores de la población a los que les sería útil obtener una forma de acceso regular a la información virtual actualizada en sitios web.

Teniendo en cuenta lo anteriormente expuesto se plantea el siguiente **problema de investigación**: ¿Cómo contribuir a la divulgación de contenidos publicados en sitios web en *Wordpress* para usuarios sin conexión a Internet?

Por tanto, el **objeto de estudio** se centra en el proceso de divulgación de la información a través de dispositivos móviles, definiendo como **campo de acción** la divulgación de la información para usuarios sin conexión a Internet.

Objetivo general: desarrollar un plugin que permita el empaquetado web de contenido en *Wordpress* y una aplicación que contribuya a la divulgación de información para usuarios sin conexión a Internet.

Objetivos específicos:

- Describir el estado del arte referente al desarrollo de herramientas para el empaquetado de contenido web.
- Diseñar las funcionalidades del *plugin* para el empaquetado web de contenido en *Wordpress* y de la aplicación para la divulgación de información para usuarios sin conexión a Internet.

- Implementar las funcionalidades del *plugin* para el empaquetado web de contenido en *Wordpress* y de la aplicación para la divulgación de información para usuarios sin conexión a Internet.
- Validar el *plugin* y la aplicación a través de pruebas de *software*.

Para guiar la investigación se definen las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos que sustentan el proceso de divulgación de la información para usuarios sin conexión a Internet?
- ¿Cuáles son las características que debe cumplir el *plugin* de empaquetado web de contenido publicado en sitios web en *Wordpress* para que permita a una aplicación de Android mostrar dicho contenido?
- ¿Cómo estructurar el proceso de desarrollo del *plugin* que facilite el empaquetado web de contenido publicado en sitios web en *Wordpress* y la aplicación propuesta, de modo que muestre dicho contenido?
- ¿Cumplirán el *plugin* y la aplicación de Android con las funcionalidades previstas?

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

- Descripción del estado del arte sobre herramientas para la divulgación de la información para usuarios sin conexión a Internet.
- Levantamiento de información para identificar los requisitos de *software*.
- Selección de las tecnologías, herramientas y estándares para la implementación de la propuesta de solución.
- Diseño de la arquitectura, selección de patrones de diseño y arquitectura.
- Implementación de la solución propuesta.
- Diseño de pruebas de *software* para validar la solución implementada.

- Validación de la solución implementada.

Los métodos de investigación empleados 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 *Wordpress* 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 *Wordpress* y también cómo se le hace llegar a los usuarios registrados en ellos.

La Entrevista se empleó en encuentros con profesores y especialistas del departamento de Servicios Informáticos para Internet del centro CIDI, la cual facilitó comprender el proceso de creación de la información y su clasificación; además, cómo se implementan los portales dentro del centro, qué tecnologías utilizan y cómo se implementa la seguridad de los portales desarrollados en el departamento.

Ver anexo N°1.

La presente investigación está estructurada en 3 capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos que ampliarán la información que se aporta en la investigación. A continuación se describe brevemente su estructura, sintetizando el contenido de cada capítulo.

Capítulo 1: “Fundamentación teórica”. 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 el lenguaje que serán utilizados en la construcción de la solución.

Capítulo 2: “Análisis y diseño del sistema”. 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, aplicando los patrones de arquitectura y diseño seleccionados.

Capítulo 3: “Implementación y prueba del sistema”. En este capítulo mediante el apoyo en los diagramas de componentes y de despliegue, se describe la herramienta propuesta desde el enfoque de la programación. Además, se realizan casos de prueba para comprobar que el *software* cumple con los requerimientos establecidos, se describe la implementación y posterior validación realizada al producto obtenido como solución.

CAPÍTULO 1: Fundamentación teórica.

En el presente capítulo se hace un análisis de los principales conceptos asociados a la investigación como: “Navegación offline” y “APK para teléfonos Android”. Se definen y describen las herramientas a utilizar, así como sus características y criterios de selección; de la metodología de desarrollo de *software* a utilizar, sus fases, flujos de trabajo y artefactos que se proveen. Se realiza un análisis de las soluciones existentes, justificando la negativa para emplear algunas de estas y definiendo características reutilizables.

1.1. Conceptos asociados al dominio del problema

¿Qué es Navegación *offline*?

Se habla de navegación *offline* cuando es posible acceder a uno o más sitios web porque ya ha sido guardada una versión copiada localmente ya sea por el navegador web o por algún programa especial para tal fin (Alegsa, 2016).

Para el Instituto de Tecnologías Educativas de España se define como el poder de consultar los sitios web sin necesidad de estar conectados a Internet y permite además garantizar la disponibilidad, a la vez que evita la dispersión y focaliza la atención en un contenido concreto ya que, al consultarlos sin estar conectados a Internet, limita las posibilidades de navegación solo a los sitios descargados (ite, 2017).

Existen diferentes conceptos y percepciones de lo que es navegación *offline*, sin embargo, para este trabajo se toma un resumen de los conceptos plasmados con el que se trabajará. Definiéndose así como navegación *offline*: el acceso a la información de un sitio web si necesidad de estar conectado a Internet, lo que trae consigo mayor rapidez a la hora de obtener la información del sitio.

¿Qué es una APK para teléfonos Android?

Al igual que las computadoras, los celulares inteligentes usan sistemas operativos que permiten a los usuarios ejecutar aplicaciones o programas además de hacer llamadas. Uno de ellos es Android, el cual está asociado con el uso de los archivos .apk. Esta clase de ficheros también se les conoce como paquetes de archivos de Android y contienen *software* para los teléfonos celulares con este sistema operativo,

además de servir como marcador para el dispositivo que permite a éste reconocer y ejecutar el programa (unocero, 2015).

Las APK para teléfonos Android son definidas de la siguiente forma:

“APK es un archivo ejecutable que empaqueta en su interior una aplicación para Android, lista para ser instalada; algo similar a los .exe en Windows o a los .dmg en Mac. Sus siglas provienen de Application Package File y son archivos para instalar aplicaciones de Android más allá de la Google Play Store” (Donoso, 2016).

“Se le llama APK a los archivos con extensión que corresponden a .apk y cuyo significado completo es Application Package File, el cual fue pensado para el sistema operativo Android” (Peña, 2015).

“Un **archivo APK** es el formato utilizado para la instalación de *software* en Android. Éste es una variante del .jar de Java y se utiliza para distribuir e instalar componentes empaquetados en smartphones o tablets” (Muñoz, 2016).

La definición más acertada para esta terminología, según la opinión del autor, acorde con la esfera informática y el desarrollo de *software*, es la siguiente: “APK es el tipo de formato que utiliza Android para la instalación de sus softwares, sus siglas provienen de *Application Package File*. Los archivos APK permiten instalar aplicaciones filtradas para acceder a nuevas características que de otra manera no serían accesibles”.

1.2. Estudio de homólogos

Existen diferentes herramientas para la distribución o consumo de información *offline*, tales como los *plugins WP HTML Static Output* y el *Simply Static* del CMS *Wordpress*. De las aplicaciones de escritorio se selecciona para su análisis *HTTrack* como una opción a explorar para el empaquetado de portales web. Además, en el ámbito nacional se estudian alternativas que hacen uso de la navegación *offline* explotando las crecientes tecnologías móviles existentes en el país.

1.2.1. Internacionales

WP HTML Static Output: salida estática *Wordpress* HTML que permite seleccionar un sitio desarrollado sobre *Wordpress* y elegir sus caminos en base a criterios para exportar a HTML. Mediante el uso de los *hooks* es capaz de añadir, definir o modificar los caminos, activos y datos del HTML utilizado en la exportación (**Stafford**, 2015).

Ventajas

- Soporta diferentes rutas para exportar entre ellas por nodos.
- Criterios de selección de *hooks* mediante las *API* de *Wordpress* o por vistas.

Desventajas

- No existe una versión estable para *Wordpress* ya que sigue en desarrollo.

Simply Static: 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*¹ (Ramon, 2010). A continuación, se describen las principales características de este módulo:

Ventajas

- 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.
- 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.

¹ Memcached es un sistema distribuido de alta performance para el cacheo de objetos en memoria, genérico por naturaleza, pero pensado para incrementar la velocidad de aplicaciones web dinámicas, aliviando la carga de las bases de datos

- La mayor parte de los puntos de fallos habituales se encuentran en consultas de PHP o SQL, las cuales ya no forman parte del sistema, por lo que hace a un sitio más confiable.

HTTrack: es una aplicación informática del tipo “*Software libre*” con Licencia GPL, multilenguaje y multiplataforma que es empleado para la captura de sitios web, es decir la descarga a un directorio local, de todo o parte de un sitio web, para poder navegar en cualquier momento sin necesidad de estar conectado a Internet (Soto, 2012).

El programa distingue entre enlaces internos y externos, adaptando los primeros y dejando intactos los segundos. 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 los descargados anteriormente y reanudar descargas interrumpidas. Es totalmente configurable y tiene un sistema de ayuda integrado (Kauler, 2015).

1.2.2. Nacionales

En el ámbito nacional existe un gran flujo de información virtual sobre todo en los últimos años. Actualmente existen sistemas de distribución “*offline*” de contenidos televisivos extranjeros, programas, música y otros productos digitales como el denominado “paquete”. Dentro de la isla se pueden destacar aplicaciones de sistemas Android como “Zapya” o “Conoce Cuba” las cuales no necesitan de conexión a Internet para brindar sus servicios, esta última brinda mapas de todas las provincias del país e información de lugares de interés para el usuario que acude al servicio.

Una herramienta para la distribución o consumo de información *offline* es el módulo “*Web Offline*” desarrollado en el CMS Drupal, el cual permite descargar un sitio desarrollado en Drupal, en una base de datos SQLite, la que es vista mediante la aplicación Android del mismo nombre; tanto el módulo como la aplicación fueron desarrollados en la Universidad de las Ciencias Informáticas. Esta aplicación permite obtener información presente en los sitios web desarrollados en Drupal y distribuirla sin conexión explotando las tecnologías móviles existentes en el país.

Ventajas

- Acepta a los usuarios la selección y configuración de la información que desean obtener.
- Admite el uso de los *hook* y nodos para crear caminos mediante los cuales se exportarán los diferentes recursos del sitio.
- Aprueba la exclusión del lado del servidor para aumentar el rendimiento, confiabilidad y seguridad.
- Obtiene la posibilidad de distribuir la información a través de aplicaciones para Android.

Resultados obtenidos del estudio

Como resultado del estudio realizado se concluye que ninguna de las aplicaciones analizadas de empaquetado web, constituye una solución factible. No existe una aplicación desarrollada en *Wordpress* que permita obtener la información presente en la web y distribuirla sin conexión explotando las tecnologías móviles existentes en el país. No obstante, se identificaron características y componentes reutilizables en el desarrollo de la solución, como el uso de los *hook* y nodos para crear caminos mediante los cuales se exportarán los diferentes recursos del sitio; la aceptación a los usuarios de la selección y configuración de la información que desean obtener de cualquier tipo de contenido (textos, imágenes, videos y música) y la exclusión del lado del servidor para aumentar el rendimiento, confiabilidad y seguridad. Por lo tanto, es necesario desarrollar una propuesta de solución que permita distribuir la información a través de una aplicación para Android, asegurando una mayor divulgación de la información descargada.

1.3. 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.

1.3.1. Metodología de desarrollo de *software*

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 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 (Pressman, 2003).

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 también denominadas 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 (Pressman, 2003). A continuación, se analizan brevemente algunas de las metodologías más utilizadas.

OpenUP

Open Unified Process o Proceso Unificado Abierto es un modelo de desarrollo de *software*. 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 diseñada para entornos dinámicos en los cuales se presentan equipos de desarrollo pequeños (hasta 10 programadores). Orientada fuertemente hacia la codificación y programación disminuye el tiempo de desarrollo de los proyectos. Crea énfasis en la comunicación informal, verbal con muy poca documentación necesitada en el desarrollo del sistema (Letelier, 2005). Su planificación es por entregas y entre sus características está priorizar aquellas *user-*

*story*² que el cliente selecciona porque son más importantes para el negocio; las entregas son lo más pequeñas posibles, se dividen en iteraciones (iteración = 2 ó 3 semanas) y están compuestas por historias de usuarios. A través de esta metodología se le asigna a cada programador una tarea de la *user-story* y la programación de tareas se realiza por parejas, donde la pareja diseña, prueba, implementa e integra el código de la tarea.

Metodología AUP-UCI

El Proceso Unificado Ágil (AUP, por sus siglas en inglés) variante UCI (AUP-UCI) surge de la necesidad de converger a una única metodología de desarrollo que cubra las particularidades de los distintos centros productivos de la UCI. Al igual que AUP (que es una versión simplificada de RUP), describe de una manera simple y fácil de comprender, la forma de desarrollar *software* utilizando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. AUP-UCI define 3 fases (inicio, ejecución, cierre), 8 disciplinas (modelado de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación, pruebas de aceptación, despliegue) y 11 roles (jefe de proyecto, planificador, analista, arquitecto de información, desarrollador, administrador de la configuración, *stakeholder* (cliente/proveedor de requisitos), administrador de calidad, probador, arquitecto de *software*, administrador de bases de datos). Los requisitos funcionales pueden ser encapsulados en historias de usuario, en descripción de requisitos por procesos o en casos de uso. Esta metodología, además de estandarizar el proceso productivo en la UCI, da cumplimiento a las buenas prácticas que define *Capability Maturity Model Integration* (CMMI, por sus siglas en inglés) v1.3. (Rodríguez, 2014).

Selección de la metodología

Para guiar el proceso de desarrollo de *software* se selecciona la metodología AUP-UCI ya que existen adaptaciones para el centro en el cual se desarrollará la aplicación. Esta metodología 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.

² Historias del Usuario establecen los requisitos del cliente que no son más que trozos de funcionalidad que aportan valor definido por el cliente.

1.3.2. 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*, así como la integración para gráficos vectoriales (SVG) y MathML para fórmulas matemáticas. Estas características permiten incluir y controlar contenidos multimedia en la web sin tener que recurrir a *plugins* ni *APIs* propietarias (Villena, 2009).

Cascading Style Sheets 3 (CSS): es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "*documentos semánticos*"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (Libros Web, 2006).

PHP 5: es un lenguaje de programación utilizado para la creación de sitios web, interpretado del lado del servidor y 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 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: es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como *node.js* o *Apache CouchDB*. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional (jsx, 2016).

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 él se pueden ejecutar en diversas plataformas con sistemas operativos como Windows, Mac OS, Android, Linux o Solaris. Los programas Java son portables, 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 unificado 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 (Popkin *Software and Systems*, 2007). 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.

Selección de Lenguajes utilizados

Se seleccionaron todos los lenguajes antes propuestos, PHP para la programación del *plugins* y todos los relacionados con la web (HTML, CSS y JavaScript) para estructurar, presentar, modelar y darle estilo al sitio después de realizar el empaquetado web.

1.3.3. Entornos de Desarrollo

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 (NetBeans Community, 2000).

Aptana Studio: es un entorno de desarrollo de código abierto basado en Eclipse. 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 y PHP. 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 (The Eclipse Foundation, 2016).

Android Studio: es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de *IntelliJ*, Android Studio ofrece funciones que aumentan la productividad durante la compilación de apps para Android, entre ellas la compilación flexible basado en *Gradle*, *emulador* rápido

con varias funciones, posee un entorno unificado en el que realizan desarrollos para todos los dispositivos Android, posee carreras instantáneas para aplicar cambios mientras las apps se ejecutan sin la necesidad de compilar una nueva *APK*, integración de plantillas de código y *GitHub* para ayudar a compilar funciones comunes de las apps e importar ejemplos de código, gran cantidad de herramientas y *frameworks* de prueba, herramientas Lint para detectar problemas de rendimiento, uso, compatibilidad de versión y compatibilidad con C++ y NDK Soporte integrado para *Google Cloud Platform*, que facilita la integración de *Google Cloud Messaging* y *App Engine* (Android Studio, 2011).

Selección del Entorno Desarrollo Integrado

Se seleccionó NetBeans en su versión 8.1 para el desarrollo de la *plugins* pues 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 útil para el desarrollo web.

Para el desarrollo de la *APK* se decide utilizar Android Studio en su versión 2.2.2 ya que es de código abierto y multiplataforma permitiendo un mayor alcance de las aplicaciones desarrolladas sobre él. Es el *IDE* oficial para Android propuesto por Google. Es compatible con la mayoría de los sistemas operativos como Windows, Mac, GNU/Linux. De modo que se convierte en una herramienta útil para el desarrollo de aplicaciones para dispositivos con sistema operativo Android.

1.3.4. Sistemas gestores de base de datos

DataBase Management System o Sistema de Gestión de Base de Datos (SGBD), es un sistema de *software* que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando su seguridad e integridad. Entre los principales gestores de Bases de Datos a nivel mundial se encuentran Oracle, SQL Server y MySQL, PostgreSQL.

MySQL: es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales, lo que permite gestionar archivos llamados de bases de datos; utiliza múltiples

tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos como Windows, Mac, GNU/Linux (esepe studio, 2005).

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 sub-consultas; 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 bases de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution*). Utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. La conexión puede ocurrir vía TCP/IP o sockets locales (Rafael, 2010).

Algunas de las características más importantes y soportadas por PostgreSQL son que permite realizar copias de seguridad en caliente (*Online/hot backups*), es unicode, posee juegos de caracteres

internacionales, permite la regionalización por columna, presenta múltiples métodos de autenticación, acceso encriptado vía SSL, actualización in-situ integrada (*pg_upgrade*), posibilita el completamiento de documentación, posee licencia BSD y *APIs* para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme y Qt.

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 *plugins* de *Wordpress* son independientes del sistema gestor de base de datos que se utilice ya que deben ser compatibles con todos los que implementen lenguaje SQL. Para el desarrollo del Sistema de empaquetado de contenido web para *Wordpress* se decide optar por MySQL en la versión 5.7.14 ya que este gestor es el que utiliza el CMS *Wordpress* 4.5.2, 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 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.

1.3.5. 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 CGI³, 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 *plugins* de *Wordpress* 4.5.2 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 2.4.23 porque en comparación con Nginx es más robusto 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 de usuarios, la visualización del código HTML en varios niveles, registra los errores en varios formatos y tiene soporte para *host* virtuales.

1.3.6. Herramientas

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar tareas de modo más sencillo. Para seleccionar las herramientas que se usarán en la solución propuesta se hace un estudio de las siguientes:

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

³ Interfaz de entrada común (en inglés *CommonGatewayInterface*).

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.

Wordpress 4.7.2: es un sistema de gestión de contenidos o CMS (por sus siglas en inglés, *Content Management System*) enfocado para crear cualquier tipo de sitio. Con el tiempo se ha convertido en una de las principales herramientas para crear páginas web comerciales. Ha sido desarrollado en el lenguaje PHP para entornos que ejecuten MySQL y Apache, bajo licencia GPL y es *software* libre. Dicho CMS presenta potencialidades que hacen de éste uno de los más utilizados a nivel mundial. Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y sus características como gestor de contenidos. Es un sistema de publicación personal distribuido libremente, basado en estándares web, rápido, ligero y gratis; con una configuración y características muy bien pensadas y un núcleo extremadamente personalizable. (Wordpress, 2013).

Herramienta CASE

Visual Paradigm8.0: 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.

PhpMyAdmin: es una herramienta de *software* libre escrita en PHP, destinada a gestionar la administración de MySQL a través de la Web. PhpMyAdmin soporta una amplia gama de operaciones en MySQL y MariaDB. Las operaciones de uso frecuente (gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios y permisos) se pueden realizar a través de la interfaz de usuario, mientras que todavía tiene la capacidad de ejecutar directamente cualquier sentencia SQL (phpmyadmin, 2003).

PhpMyAdmin 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 MySQL y facilita la administración. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados (phpmyadmin, 2003).

Herramientas para pruebas del sistema

En la realización de las pruebas de rendimiento al *plugin* de empaquetado web de contenidos para *Wordpress* 4.7.2 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).

1.4. 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 *plugin* para *Wordpress* 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 *plugin* haciendo uso de cualquier Gestor de Base de Datos y Servidor Web por las facilidades que brinda el CMS *Wordpress*.

CAPÍTULO 2: Análisis y diseño del sistema.

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 vinculados con el entorno. También se definen las principales características que debe cumplir el sistema en términos de requisitos funcionales y no funcionales. Además 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 las 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).

De manera general se puede afirmar 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.

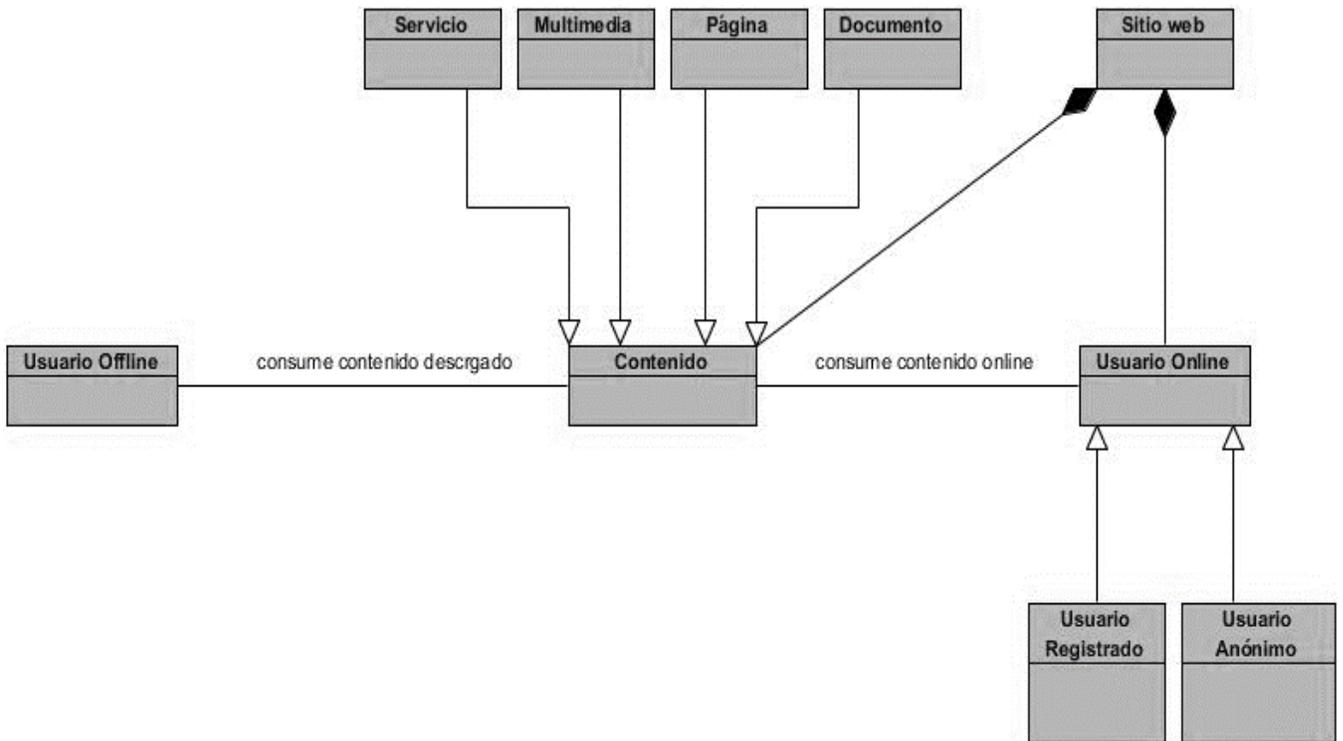


Figura 1. Diagrama de Clases del Modelo de Dominio

2.2.1. 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.

Usuario Online: es la persona que interactúa con el portal web mediante la red.

Usuario Anónimo: se refiere al usuario que no presenta una identificación previa.

Usuario 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.

Multimedia: representación visual de objetos o formas de manera digital (imagen, audio y video).

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, multimedia, 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 se define lo que debe hacer el sistema. 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.

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 *plugin* de empaquetado web de contenido.

Tabla 1. Requisitos Funcionales.

Código	Descripción (Requisitos Funcionales) del plugin (WP2SQLite)	Prioridad
RF1	Modificar Título.	Baja
RF2	Seleccionar Tipo de Contenido.	Alta
RF3	Seleccionar Campos por Tipo de Contenido.	Alta
RF4	Generar Base de Datos para APK.	Alta
RF5	Descargar archivos multimedia.	Alta
Código	Descripción (Requisitos Funcionales) de la Aplicación (Wordpress contigo)	Prioridad
RF6	Consumir Base de Datos generada.	Alta
RF7	Mostrar listado de contenidos.	Media
RF8	Mostrar contenido.	Alta
RF9	Mostrar menú de navegación.	Media
RF10	Filtrar por tipos de categoría.	Alta
RF11	Buscar contenido.	Alta
RnF (Requisitos no Funcionales) para el plugin (WP2SQLite)		
	Accesibilidad	
RnF-1	Se asignarán los permisos de acceso en dependencia del rol que desempeñe cada usuario del sistema.	
	Eficiencia	

RnF-2	El sistema debe ser capaz de procesar 10 peticiones por segundo para un total de 20 usuarios concurrentes.
RnF-3	El sistema debe tener un tiempo de respuesta inferior a 5 segundos.
Restricciones del Diseño	
RnF-4	Las páginas generadas deben cumplir con los estándares para XHTML 2.0 y CSS 3.0.
RnF-5	El <i>software</i> , bibliotecas o componentes empleados deben estar regidos en la medida de lo posible por licencias y patentes de <i>software</i> libre.
Interfaz	
RnF-6	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-7	Se debe utilizar la licencia PHP License.
RnF-8	Se debe utilizar la licencia BSD de MySQL.
RnF-9	Se debe utilizar la licencia GNU/GPL para el CMS <i>Wordpress</i> .
Estándares aplicables	
RnF-10	Se sigue el estándar de programación sugerido por los desarrolladores del CMS <i>Wordpress</i> .
RnF (Requisitos no Funcionales) para la aplicación (Wordpress contigo)	
Usabilidad	
RnF-11	La aplicación debe de ser fácil de descargar e instalar.
RnF-12	Las interfaces de la aplicación deben presentar una vista limpia y de fácil navegación.

Hardware	
RnF-13	La aplicación necesita al menos 2 megas de almacenamiento para ser instalada y funcione correctamente.
RnF-14	La aplicación requiere como mínimo sistemas operativos superiores a Android 4.1.
Estándares aplicables	
RnF-15	Se siguen el estándar de programación sugerido por los desarrolladores de Java.

2.3.2. Historias de Usuario

Las Historias de Usuario (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 suficientes detalles 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 esencial para los clientes en 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. Mostrar contenido.

Tabla 2. Descripción de HU: Mostrar contenido.

Historia de usuario	
Número: HU_1	Nombre Historia de Usuario: Seleccionar campos por tipo de contenido
Programador: Rolando Vázquez Conyedo	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 almacenar en la base de datos.	
Observaciones: el campo contenido esta seleccionado por defecto, además se puede marcar los campos restantes por lo que se quiere exportar ya sea imagen, audio o video.	

Prototipo de interfaz:

DESCARGAR CONTENIDO

- Contenido
- Imágenes
- Audio
- Videos



2.4. Arquitectura

El sistema implementado sobre el CMS *Wordpress* hereda la arquitectura de este. El mismo contiene la base de datos MySQL, la cual interactúa con los *plugins* que deben ser cargados en el servidor antes de ser utilizado, este *plugin* se encuentra en un archivo de biblioteca, durante la interacción con la plantilla de contenidos desarrollados en PHP, CSS, JavaScript que sirve para dar una mejor visualización al usuario. (Figura 2).

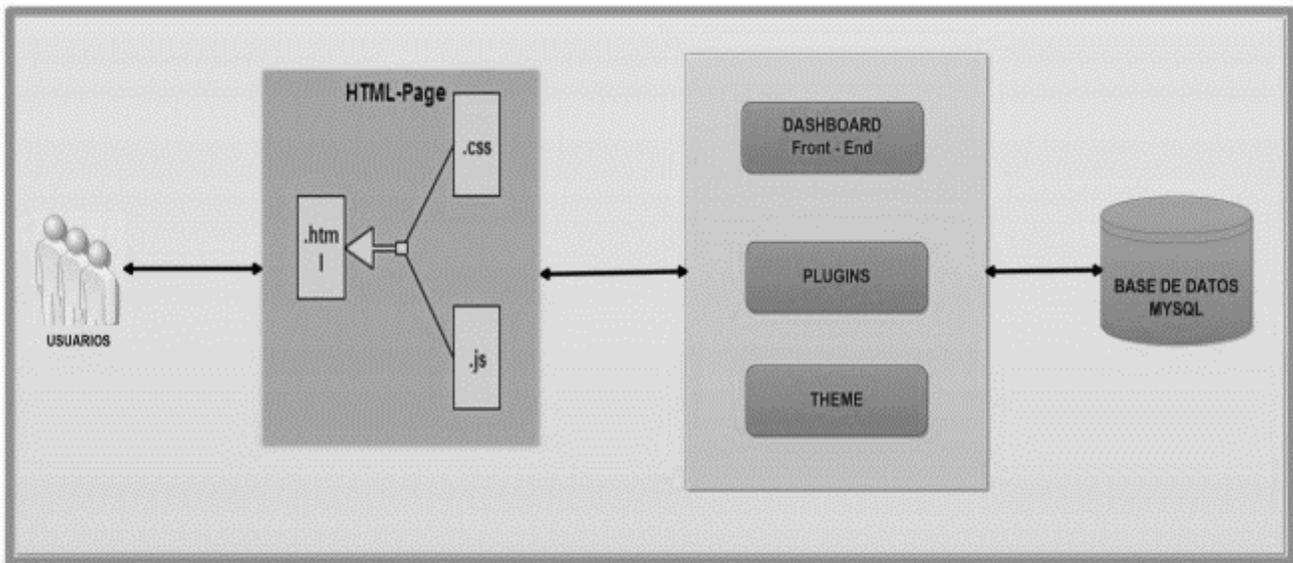


Figura 2. Arquitectura de Wordpress (Morante Arreaga, 2015).

Páginas HTML (HTML- PAGE): este componente de la arquitectura de *Wordpress* es el proceso de visualización con el cual se puede verificar que está constituido por páginas HTML, las cuales son desarrolladas mediante los lenguajes CSS y JavaScript. Se evidencia en las diversas vistas presentadas por el *plugin* durante el proceso de generación de la base de datos.

Hoja de estilos (STYLE SHEETS): amplían la capacidad de mejorar el aspecto de las páginas, es desarrollado en CSS. Se utiliza para dar un mejor acabado en las diversas vistas presentadas por el *plugin* durante el proceso de generación de la base de datos.

Guiones (SCRIPTS): son todas las líneas de código que se puedan crear en JavaScript.

Tablero (DASHBOARD): este componente es el tablero que permite proporcionar datos relativos al número de mensajes, etiquetas, comentarios, páginas, categorías a través de esto puede explorar internamente el blog creado. En el desarrollo del *plugin* de empaquetado web de contenido para *Wordpress* 4.7.2 se decide trabajar con los *posts* almacenados en la base de datos, permitiendo al usuario definir qué *posts* desea obtener y exportar.

Plantillas de contenido (CONTENT TEMPLATE): permiten realizar la configuración del tema, de manera que pueda insertar un diseño específico a los estilos del tema creado. En el caso del *plugin* 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 de formulario (FORM- TEMPLATES): a través del formulario se introduce datos en las tablas de forma ordenada. En el desarrollo del *plugin* no se insertan nuevas entradas, sino que consume las encontradas en el sitio para ser exportadas según las razones de selección del usuario.

(PLUGINS): son contenedores que permiten realizar la modificación de las bitácoras y el diseño del sitio web por medio del script PHP. Se puede agregar más funcionalidades a través de las etiquetas de plantillas con la función PHP. En esta capa se encuentra el *plugin* desarrollado permitiendo ver los cambios en los temas.

Tema (THEME): se puede crear mediante etiquetas de plantilla para mostrar ciertos tipos de contenido, como blogs, calendarios, menús de navegación, contenido de la página, vínculos de archivo, toda la funcionalidad del sitio de *wordpress* está representada por archivos .PHP.

REGIÓN TEMPLATES los tipos de contenidos que están definidos en una página web, son clases que contienen el *header*, *sidebar*, *footer* en lenguaje PHP:

- *Header* se utiliza para enviar encabezados HTTP sin formato.
- *Sidebar* funciona como barra lateral donde cargan los widget.
- *Footer* cumple una función parecida al header, porque permite cargar scripts y archivos de los *plugins*.

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.

2.4.1. 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:

Para el desarrollo del plugin (WP2SQLite).

Observer (Observador): el patrón observador es generalizado en *Wordpress*, evidenciándose en los *plugins*, 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* `register_WP2SQLite_Widget`, este registra la clase `WP2SQLite_Widget`.

Chain of Responsibility (Cadena de Responsabilidad): el sistema de menú de *Wordpress* sigue el patrón Cadena de Responsabilidad. En cada solicitud de la página, el menú del sistema determina si hay un *plugin* para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y qué 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 *plugin* atienda la petición o hasta que un *plugin* deniegue el acceso. Este patrón se puede ver evidenciado cuando se va a acceder al *plugin*, es necesario ir al *dashboard* y luego `WP2SQLite`, estos dos menús se aseguran de verificar que el usuario tenga acceso al contenido solicitado.

Bridge (puente): la capa de abstracción de datos de *Wordpress* se encuentra implementada siguiendo el patrón puente. Los *plugins* 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 `putData()` donde se hacen consultas de selección e inserción a la base de datos abstrayéndose del SGBD que se emplee para manejar los datos almacenados.

Para el desarrollo de la Aplicación Android (Wordpress contigo).

Modelo Vista Controlador (MVC): es un patrón de arquitectura de *software* que separa una aplicación en tres componentes o partes diferenciadas: el Modelo, que representa los datos de una aplicación; la Vista y el Controlador, normalmente se corresponden con la interfaz de usuario; el Controlador incluye a menudo la lógica que procesa las acciones y eventos generados durante el tiempo de vida de la aplicación; de manera que entre estos componentes existe lo que se llama "bajo acoplamiento", es decir, que la definición y detalles de cada uno de ellos (y en el caso de la Vista, hasta su propia existencia) son más o menos desconocidos para los otros dos (Ilián, 2013).

2.5. Diseño

2.5.1. 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 *Wordpress*, específicamente los paquetes que este presenta. En el siguiente diagrama se muestran, con su descripción correspondiente:

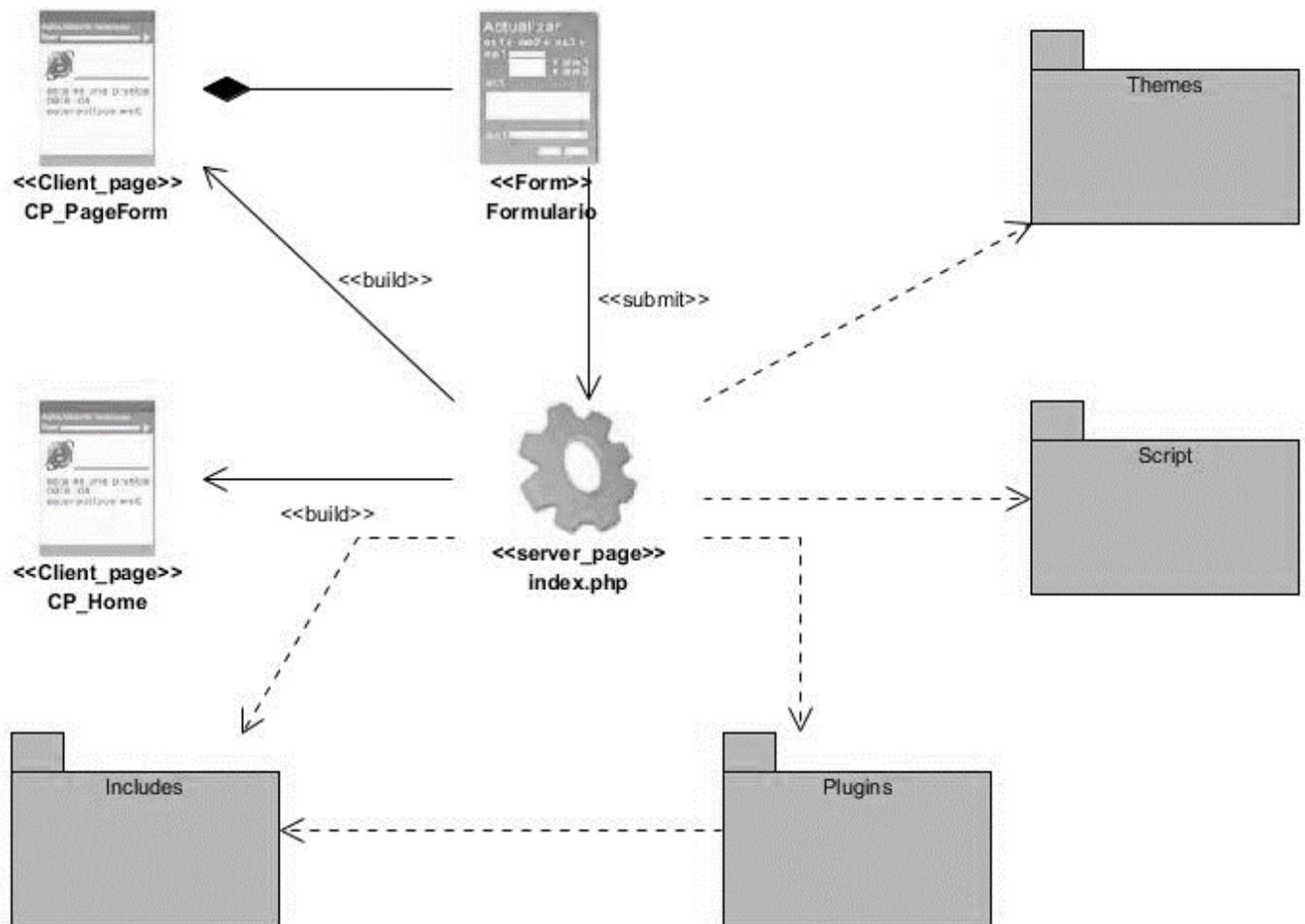


Figura 3. Diagrama de Clases de Diseño de Wordpress.

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

Plugins: espacio en el que se encuentran todos los *plugins* 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 Tipo de contenido. A continuación, se describen los distintos escenarios:

En el caso de Seleccionar Tipo de Contenido la página servidora SP_wp2sqlite construye la página cliente CP_widget, que se compone por el formulario Frm_Seleccionar_Tipo_de_Contenido, la cual 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 SP_download.

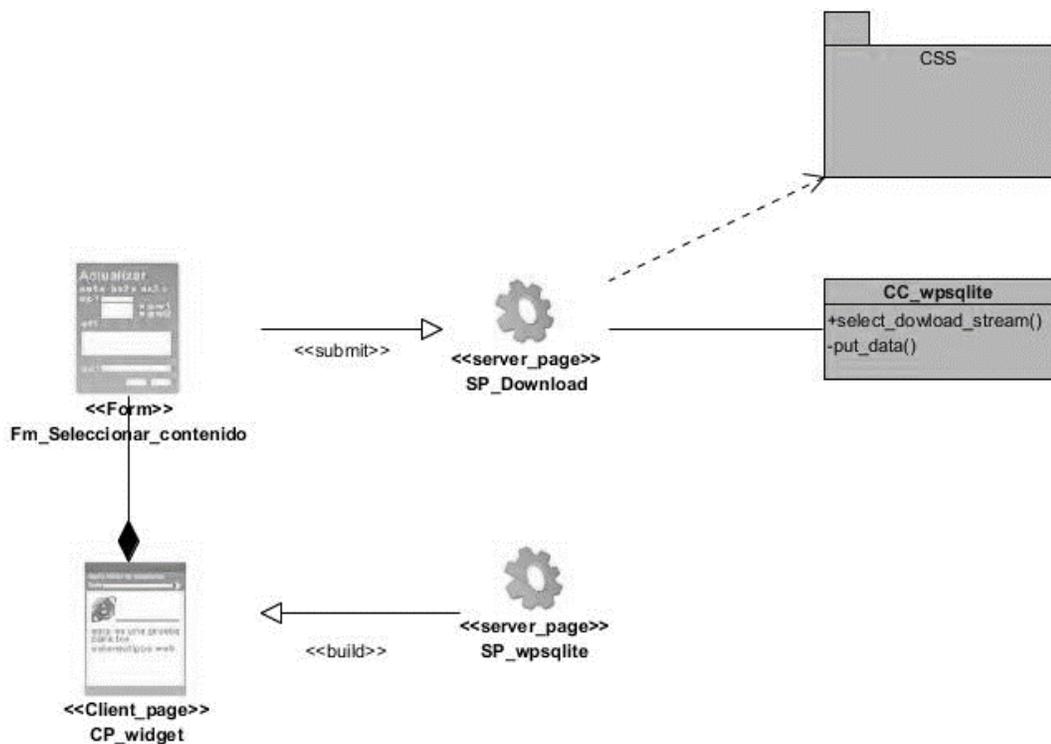


Figura 4. Diagrama de clases de diseño. Seleccionar Campos por Tipos de Contenido.

2.6. 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.

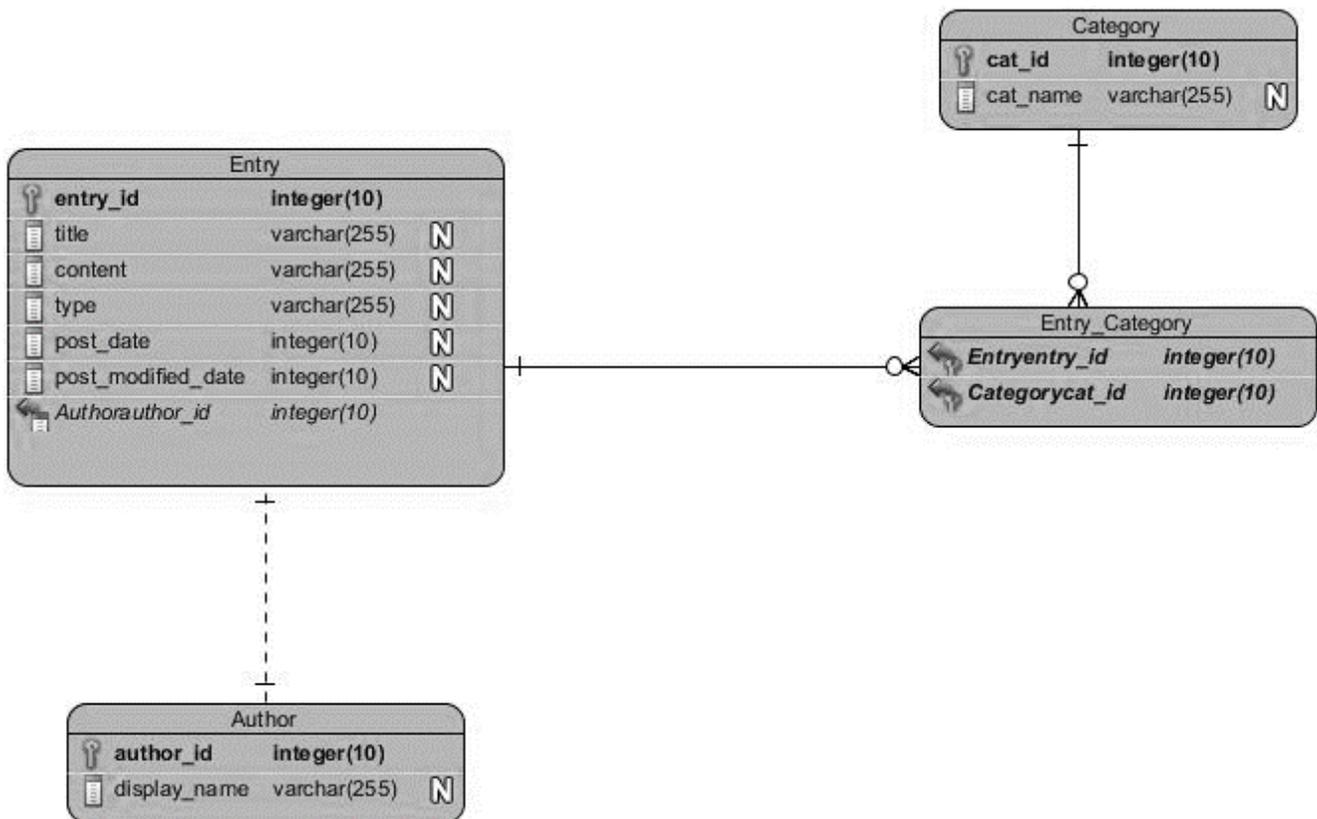


Figura 5. Modelo relacional.

2.7. 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 los *posts* y las relaciones entre ellos, los *posts* 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 SQLite o MySQL representado por el nodo “**Servidor de BD**”.

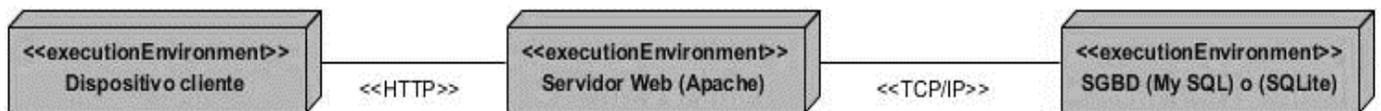


Figura 6. Diagrama de Despliegue.

2.8. Conclusiones parciales

Con la culminación de este capítulo se puede afirmar 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, especificados y 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 *plugin* de empaquetado web de contenidos para *Wordpress* 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 su relación, así como las funcionalidades y atributos que deben presentar cada una de estas. Con el modelo de despliegue se posibilitó modelar una vista de las topologías del *hardware* sobre el que se ejecuta el sistema.

CAPÍTULO 3: Implementación y prueba del sistema.

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 cómo los elementos del modelo de diseño se implementan en términos de componentes. Se realizan pruebas de caja negra al *plugin* desarrollado y pruebas unitarias a la apk implementada 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.

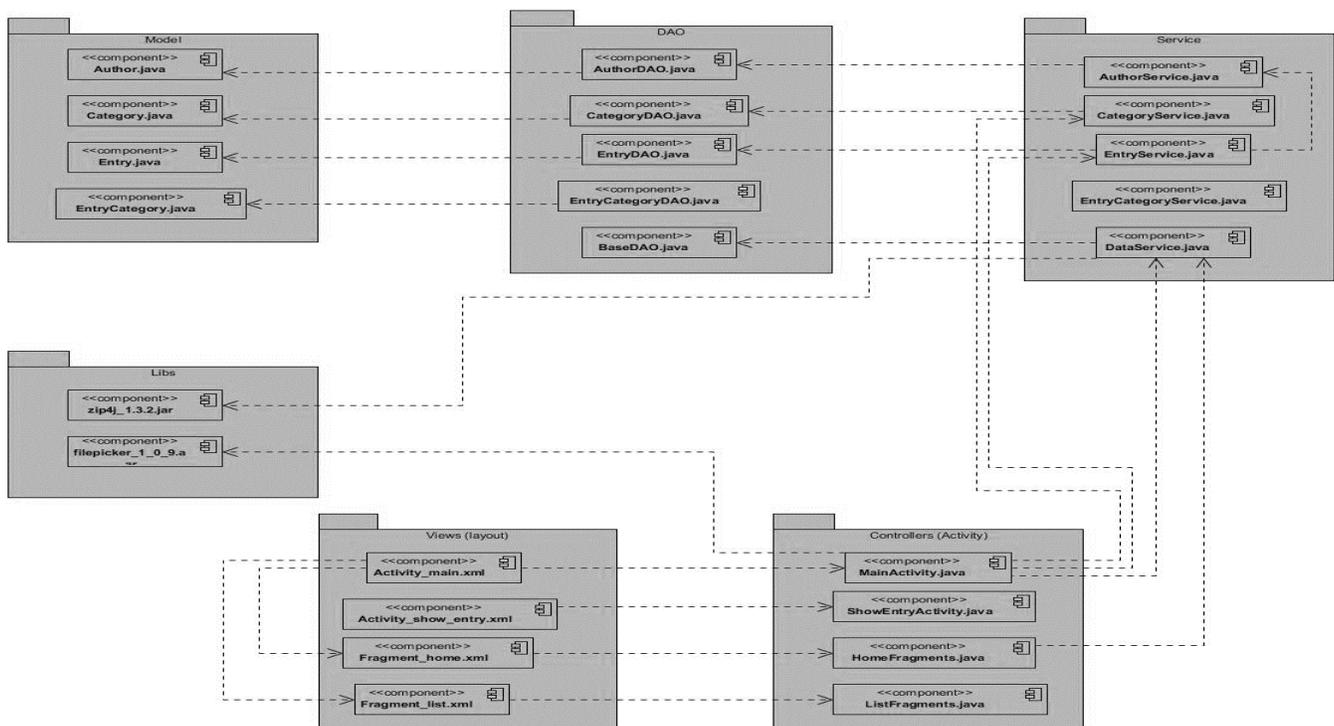


Figura 7. Diagrama de Componentes de la Aplicación.

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

Componentes	Descripción
Author	Contiene los datos de los actores que se encuentran en la base de datos <code>sqlite(id, author_id, display_name)</code> .
Category	Contiene los datos relativos a las categorías que se encuentran en la base de datos <code>sqlite(id, cat_id, cat_name)</code> .
Entry	Contiene los datos relativos a las entradas que se encuentran en la base de datos <code>sqlite(id, entry_id, title, content, author_id, type, post date, post modified date)</code> .
EntryCategory	Contiene el identificador de la entrada y el identificador de la categoría (<code>id, entry_id, cat_id</code>).
AuthorDAO	Permite el acceso a los datos del autor que se encuentran en la base de datos.
CategoryDAO	Permite el acceso a los datos de las categorías que se encuentran en la base de datos.
EntryDAO	Permite el acceso a los datos de las entradas que se encuentran en la base de datos.
AuthorDAO	Llama a la clase AuthorDAO cuando necesita los datos.
EntryCategoryDAO	Permite el acceso a los datos del autor que se encuentran en la base de datos.
BaseDAO	Es la base de datos generada por el plugin.

CategoryService	Llama a la clase CategoryDAO cuando necesita los datos.
EntryService	Llama a la clase EntryDAO cuando necesita los datos.
EntryCategoryService	
DataService	Llama a la clase BaseDAO y utiliza la librería zip4j_1.3.2 para cargar los datos.
MainActivity	Controladora de la vista principal e interactúa con los servicios (DataService, EntryService, CategoryService).
ShowEntryActivity	Controladora de la vista secundaria Show_Entry_Activity, utiliza datos de la clase DataService.
HomeFragment	Controlador del Fragment_Home.
ListFragment	Controlador del Fragment_List.
Activity_Main	Vista principal.
Show_Entry_Activity	Vista secundaria.
Fragment_Home	Muestra el contenido (por páginas o entrada).
Fragment_List	Muestra el listado del contenido (por páginas o entrada).

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 *Wordpress*, utilizados durante el desarrollo del plugin (*WP2SQLite*).

Tabla 3. Estándar de codificación del Wordpress.

Estándar de codificación	Descripción
PHP	
Estilos en Bloques	<p>Llaves de partida deben colocarse al inicio y al final de las líneas.</p> <pre> if (condition) { // Do Something } elseif (condition) { // Do Something } else { // Do Something } </pre>
HTML	
HTML	<p>HTML posee reglas asociadas, Entre ellas:</p> <ul style="list-style-type: none"> • El código debe validar contra el validador del W3C. • Las etiquetas HTML de cierre automático deben tener exactamente un espacio antes de la barra diagonal (es una especificación del W3C)
CSS	

<p>Selectores</p>	<p>Los selectores deben ser tan cualificados como sea necesario, ser humanamente legible, estar en minúsculas con palabras separadas con guiones y selectores de atributos deben utilizar comillas dobles. He aquí un ejemplo:</p> <pre>input[type="text"], input[type="password"], .name-field { background: #f1f1f1; }</pre>
<p>Javascript</p>	
<p>Punto y Coma</p>	<p>Punto y coma al final de cada sentencia valida.</p> <pre>background: #f1f1f1;</pre>

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. 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, 2000). Los parámetros de las pruebas son:

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

3.3.1. Funcionales (Para el plugin y la aplicación android)

Para la realización de estas pruebas se utilizó la técnica de caja negra, realizadas desde el punto de vista de la interfaz de usuario, lo que permitió valorar el funcionamiento del *plugin*.

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

Condiciones de ejecución: el usuario debe estar autenticado.

Tabla 4. Caso de pruebas: Escenario Mostrar Contenido.

Escenario	Descripción	Manual de configuración	Respuesta del sistema	Flujo Central
EC 3.1 Mostrar listado de contenido	El sistema muestra un listado por título de todo el contenido que se encuentra en la base de datos	V	El sistema verifica si se mostró el listado de contenido	El usuario puede ver el listado del contenido, para marcar el que quiere ver.
		Al menos un valor es modificado		
EC 3.2 Mostrar listado de contenido	El sistema muestra un listado por título de todo el contenido que se encuentra en la base de datos	I	El sistema verifica si se mostró el listado de contenido	
		Ningún valor es modificado		
EC 3.1 Mostrar tipo de contenido.	El sistema muestra el tipo de contenido que se encuentra en la base de datos, que el usuario quiere ver.	V	El sistema verifica que fue seleccionado el tipo de contenido que se encuentra en la base de datos, que el usuario quiere ver.	El usuario puede ver el tipo de contenido que marcó
		Al menos un valor seleccionado		

Resultados de las pruebas funcionales para el plugin

Se probaron todas las funcionalidades que responden a los casos de pruebas del *Plugin* y se obtuvieron un total de 23 no conformidades las cuales fueron todas resueltas, siendo los principales errores detectados, ortográficos, de funcionalidad y textos en idioma inglés.

La Figura 8 desglosa las no conformidades detectadas en las tres iteraciones realizadas. En una primera iteración se recoge un total de 10 no conformidades de las cuales se resolvieron 6 y las restantes 4 se quedaron pendientes para la segunda iteración la cual arrojó un total de 8 no conformidades, resolviéndose un total de 7 quedando 5 pendiente, para la tercera iteración se detectaron 0 nuevas no conformidades y se solucionaron las 5 pendiente que quedaban de la iteración anterior.

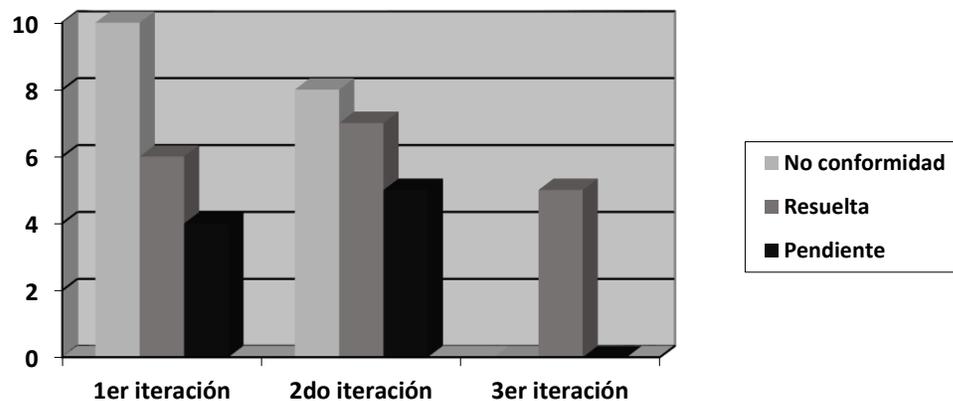


Figura 8. Resultados de las pruebas funcionales para el plugin.

Resultados de las pruebas funcionales para la aplicación

Se evaluaron todas las funcionalidades que responden a los casos de pruebas de la aplicación Android y se obtuvieron un total de 12 no conformidades las cuales fueron todas resueltas, siendo los principales errores detectados, ortográficos, de funcionalidad y textos en idioma inglés.

La Figura 9 desglosa las no conformidades detectadas en las tres iteraciones realizadas. En una primera iteración se recoge un total de 7 no conformidades de las cuales se resolvieron 5 y las restantes 2 se quedaron pendientes para la segunda iteración la cual arrojó un total de 4 no conformidades, resolviéndose un total de 3 quedando 1 pendiente, para la tercera iteración se detectaron 0 nuevas no conformidades y se solucionó la pendiente que quedaba de la iteración anterior.

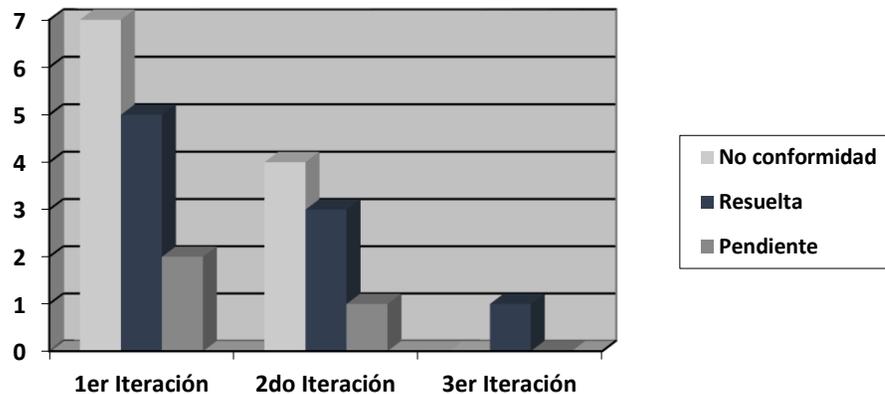


figura 9. Resultados de las pruebas funcionales para el plugin.

3.3.2. Carga y Estrés (para el plugin)

Para la realización de esta prueba se utiliza la herramienta apache-jmeter-3.1 para probar el rendimiento en los recursos, 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.

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 *plugin*, 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 *plugin*, 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 2.50 GHz

Memoria: 4GB RAM.

Tipo de Red: Ethernet 10/100Mbps.

Software:

Tipo de Servidor Web: Apache v2.4.23.

Máximo de hilos concurrentes (simulación de usuarios): 20

Plataforma: Sistema operativo *Windows* 10.

Servidor de BD: MySQL 5.7.14.

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.

Usuario	Muestras	% Error	Tiempo de respuesta promedio
20	40	0,00%	2,8 seg

Resultados de la Prueba de Carga y Estrés.

El *plugin* solo podrá ser utilizado por usuarios administradores de los sitios web o por un usuario 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 5 segundos en cada caso, el máximo de tiempo que se demora en responder la petición de un usuario es de 2.1 segundos, quedando una prueba satisfactoria.

3.3.3. Unitarias (para la aplicación android)

Las pruebas unitarias son un método para probar unidades estructurales de código. Buscan aislar cada parte del programa y mostrar que las partes individuales son correctas. Facilitan la reestructuración del código a la vez que reducen la cantidad de errores y el tiempo en localizarlos. Las propias pruebas pueden considerarse documentación porque son una implementación de referencia de cómo utilizar el código (Sommariva, 2014).

Seguidamente se describe la forma en que se realizará el proceso de pruebas unitarias:

Se crearán las clases y métodos necesarios para probar cada Tarea de implementación generada. En caso de detectar errores en el funcionamiento de alguna de las Tareas de implementación, se realizarán los cambios pertinentes en el código fuente y se le volverá a someter dicha tarea a nuevas pruebas unitarias. Esto implica que a cada Tarea de implementación se le realizará al menos una iteración de pruebas de este tipo.

Se realizaron tres iteraciones de pruebas unitarias utilizando *Android Testing Framework*. A continuación, se muestran las no conformidades detectadas durante su realización.

Iteración #1

Se realizó a las HU Mostrar contenido y mostrar listado de contenido.

- Excepción al juzgar una respuesta vacía.
- Excepción al cifrar la respuesta del usuario.
- Al introducir la respuesta correcta a un problema se obtuvo un veredicto negativo.
- Al listar los contenidos se obtuvo una lista vacía.

Iteración #2

Se realizó a las HU Buscar contenido y Filtrar por tipo de categoría.

- Al buscar contenido con el patrón “a” se obtuvo una lista con 2 tipos de contenido cuando deberían haber sido 5.
- Al filtrar por tipo de categoría por “Deporte” se obtuvo una lista que contenía categoría Deporte y sin categorías.

Iteración #3

No se encontraron no conformidades.

Las no conformidades detectadas durante la realización de las pruebas unitarias fueron corregidas satisfactoriamente.

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 de la aplicación *Wordpress*

contigo que consumiera la base de datos generada por el *plugin WP2SQLite*, basándose en el análisis y diseño de los capítulos anteriores. Además, para validar la respuesta a los requerimientos identificados se realizaron pruebas funcionales de caja negra al *plugin* y a la apk además de pruebas unitarias y de aceptación. 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 *Wordpress*, 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, las pruebas de aceptación conjuntamente con el cliente, detectaron y solucionaron errores en el proceso de desarrollo y las pruebas unitarias realizadas al código de la aplicación definidas por los desarrolladores, permitió que se lograra finalmente una aplicación que responde al conjunto de funcionalidades identificadas.

CONCLUSIONES

Una vez finalizada la fundamentación teórica que sustenta la presente investigación, definidas las características del *plugin* y la aplicación, efectuado su desarrollo y validación, se obtuvieron resultados que permiten arribar a las siguientes conclusiones:

- Con el estudio del estado del arte logró identificarse un conjunto de herramientas y tecnologías para la proyección e implementación de la presente solución informática.
- El diseño e implementación de las funcionalidades del *plugin* y la aplicación contribuyeron a la propuesta de solución, posibilitando un mayor soporte a la implementación de los requisitos identificados; logrando una mayor organización y estructura del código fuente, disminuyendo el impacto ante futuras modificaciones.
- La validación de las funcionalidades del *plugin* y la aplicación permitió realizar pruebas de *software* para 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 *plugin* (WP2SQLite) que permite el empaquetado del contenido presente en sitios web desarrollados en *Wordpress* 4.7.2, en conjunto a una aplicación Android (*Wordpress contigo*) que lee y muestra el contenido de estos paquetes para 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 (Wordpress contigo) conectarse a un portal web desarrollado sobre *Wordpress* 4.7.2 y descargar o actualizar la base de datos necesitada.

REFERENCIAS

- Acunetix. 2016.***acunetix*. [En línea] Acunetix, 2016. [Citado el: 9 de noviembre de 2016.]
- Android Studio. 2011.***Android Studio*. Conoce Android Studio [En línea] 2011. [Citado el: 15 de noviembre de 2016.] <https://developer.android.com/studio/intro/index.html?hl=es-419>.
- Alegsa, Leandro. 2016.** Definición de Offline o fuera de línea (informática). [En línea] 27 de junio del 2016. [Citado el: 12 de enero de 2017] <http://www.alegsa.com.ar/Dic/offline.php>
- Álvarez, Miguel Ángel. 2011.***DesarrolloWeb*. [En línea] 2011. [Citado el: 2 de noviembre de 2016.] <http://desarrolloweb.com/>.
- esepe studio. 2005.***esepestudio*. ¿Qué es MySQL? [En línea] martes, 16 de agosto de 2005. [Citado el: 1 de diciembre de 2016.] <http://www.esepestudio.com/noticias/que-es-mysql>.
- ETECSA. 2017.** Acceso público a Internet. *etecsa.cu*. [En línea] 2016. [Citado el: 1 de febrero de 2017.] http://www.etecsa.cu/Internet_conectividad/Internet/.
- Estevan, Gomez, Ricardo. 2016.** *espe.edu.ec*. Arquitectura de Android [En línea] 31 de mayo de 2016. [Citado el: 27 de marzo de 2017.] <https://sites.google.com/a/espe.edu.ec/programacion-ii/android/a-1---arquitectura-de-android>.
- Fawkes, Guy. 2017.** *vpnmentor.com*. Tendencias de Internet, estadísticas y datos en los Estados Unidos y el mundo para 2017 [En línea] 1 de enero de 2017. [Citado el: 24 de abril de 2017.] <https://es.vpnmentor.com/blog/page/2/>.
- Foundation, Apache Software. 2011.** Apache. *Apache JMeter*. [En línea] 09 de mayo de 2011. [Citado el: 9 de noviembre de 2016.] <http://jmeter.apache.org/>.
- Gama y autores. 2003.***Reutilización de software*. Patrones de diseño. <http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf>
- Guevara, Jorge Martínez Ladrón de. 2012.***Fundamentos de programación en Java*. Madrid: EME, 2012.
- Inc Nginx. 2011.** NGINX. *neging.org*. [En línea] 2011. [Citado el: 8 de noviembre de 2016.] <https://www.nginx.com/>.

Ite. 2017. *Educacion.es*. Navegación offline. [En línea] 2017. [Citado el: 27 de febrero de 2017.] http://www.ite.educacion.es/formacion/materiales/157/cd/m2_1_navegacion/navegacin_offline.html.

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

jsx. 2016. *Mozilla*. Java script. [En línea] 2016. [Citado el: 2 de noviembre de 2016.] <https://developer.mozilla.org/es/docs/Web/JavaScript>.

Kauler, Leto. 2015. HTTrack website copier. [En línea] 2015. [Citado el: 27 de octubre de 2016.] <https://www.httrack.com/>.

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

Letelier, Patricio. 2005. *Metodologías Ágiles y XP*. Valencia: Departamento de Sistemas Informáticos y Computación, 2005.

LibrosWeb, 2006. *¿Qué es CSS?* s.l. : LibrosWeb.es, 2006. http://librosweb.es/libro/css/capitulo_1.html.

Ilián García, Raúl. *Diseño e implementación de un juego Android para soporte a la enseñanza de frameworks OO y patrones de diseño*. 2013.

Luca, Juan Ignacio de Tena. 2004. *Manual Imprescindible de PHP5*. Madrid: GRUPO ANAYA, S.A., 2004.

Marin, Marisela. 2012. *Búsqueda de información*. [En línea] 25 de junio del 2012. [Citado el: 17 de enero de 2017.] <http://pasoapasoatusalud.blogspot.com/>

Miani, Antonio Aliaga y Marco. 2011. I.E.Sanvicente. *PostgreSQL*. [En línea] 2011. [Citado el: 8 de noviembre de 2016.] <http://www.iessanvicente.com/colaboraciones/postgreSQL.pdf>.

Montiel, Daniel Ponsoda. 2008. *Introducción a SQLite*. s.l.: Creative Commons. 2008.

Moreno, Manuel. 2017. *Trecebits*. Mapa mundial del acceso a Internet. [En línea] 26 de enero de 2017. [Citado el: 31 de enero de 2017.] <http://www.trecebits.com/2017/01/26/mapa-mundial-del-acceso-a-Internet/>

- Muñoz de Frutos, 2016.** computerhoy.com. *¿Qué es APK?* [En línea] 8 de octubre de 2016. [Citado el: 10 de febrero de 2017.] <http://computerhoy.com/noticias/moviles/que-es-apk-52222>
- Naramore, Elizabeth. 2005.** *Beginning PHP5, Apache, and MySQL Web Development*. Canadá: Wiley Publishing, 2005.
- Narom, José Luis. 2013.** *Narom. Cuántas paginas web hay en Internet 2013* [En línea] 2013. [Citado el: 31 de enero de 2016.] <http://www.narom.org/Cuantas%20paginas%20web%20hay%20en%20Internet%202013.html>
- NetBeans Community. 2000.** NetBeans Community. [En línea] 2000. [Citado el: 2 de noviembre de 2016.] <http://netbeans.org/>.
- OMG. 2015.** OMG. *Introduction to OMG's*. [En línea] 24 de junio de 2015. [Citado el: 2 de noviembre de 2016.] http://www.omg.org/gettingstarted/what_is_uml.htm.
- phpmyadmin. 2003.** *phpmyadmin*. [En línea] 2003 a 2016. [Citado el: 9 de noviembre de 2016.] <https://www.phpmyadmin.net/>.
- Peña. 2015.** *Paréntesis.com. ¿Para qué sirve?* [En línea] 8 de abril de 2015. [Citado el: 27 de febrero de 2017.] http://parentesis.com/tutoriales/Que_es_un_APK_y_como_se_instala.
- Popkin Software and Systems. 2007.** tldp. [En línea] 19 de 6 de 2007. [Citado el: 2 de noviembre de 2016.] <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.
- Rafael, Martínez. 2010.** *postgresql*. Sobre PosgreSQL. [En línea] 2 de octubre del 2010. [Citado el: 1 de diciembre de 2016.] http://www.postgresql.org.es/sobre_postgresql.
- Ramon, Jose. 2010.** *forobeta*. Simply Static - nuevo plugin, todos tus archivos "estáticos". [En línea] 14 de febrero del 2010. [Citado el: 27 de octubre de 2016.] <http://forobeta.com/plugins-wp/441107-simply-static-nuevo-plugin-todos-tus-archivos-estaticos.html>.

Redes sociales. 2016. *Franbarquilla*. ESTUDIO SOBRE EL ESTADO DE INTERNET Y LAS REDES SOCIALES EN EL 2016 [En línea] 2 de febrero de 2016. [Citado el: 31 de febrero de 2017.] <http://franbarquilla.com/estudio-sobre-el-estado-de-Internet-y-las-redes-sociales-en-2016/>.

Rodríguez Sánchez, Tamara. *Metodología de desarrollo para la Actividad productiva de la UCI*. 2014.

Robertson, James. 2008. Step Two Designs Pty Ltd. [En línea] 2008. [Citado el: 9 de noviembre de 2016.] http://www.steptwo.com.au/papers/kmc_what.

Rosa Donoso. 2016. *Androidpit*. ¿Qué es APK? [En línea] julio 2016. [Citado el: 1 de diciembre de 2016.] <http://www.androidpit.es/android-para-principiantes-apk>.

Stafford, Leon. 2015. *Wordpress*. WP Static HTML Output. [En línea] 2015. [Citado el: 27 de octubre de 2016.] <https://wordpress.org/search/site/HTML%20Export>.

Sommariva, Andrés. 2014. *Pruebas Unitarias, Parte 1: Introducción y utilización de objetos simulados (Mock)*. [En línea] [Citado el: 26 de Abril de 2016.] [53] <http://www.microgestion.com/index.php/mg-developers/articulos/74-unit-test-part1-mock>.

Soto, Karen C. Zerquera. 2012. *revista.jovenclub.cu*. HTTPTrack WebSite Copier 3.44-1 (WebHTTPTrack). [En línea] 14 de marzo del 2012. [Citado el: 27 de octubre de 2016.] <http://revista.jovenclub.cu/httptrack-website-copier-3-44-1-webhttptrack/>.

The Eclipse Foundation. 2016. *Eclipse*. *About the Eclipse Foundation*. [En línea] 2016. [Citado el: 2 de noviembre de 2016.] <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: 9 de noviembre de 2016.] <http://www.slideshare.net/StalinTuza/ingenieriasoftwareopenup#btnPrevious>.

UIT. 2015. Unión Internacional de Telecomunicaciones. *Measuring the Information Society Report*. [En línea] 2015. [Citado el: 12 de enero de 2017.]

https://www.itu.int/en/ITU-D/Statistics/Documents/publications/mis2014/MIS2014_without_Annex_4.pdf.

unicero. 2015. Unicero.com. ¿Qué es un archivo apk y como se instala?. . [En línea] 2015. [Citado el: 12 de enero de 2017.] <https://www.unocero.com/2015/09/03/que-es-un-archivo-apk-y-como-se-instala/>

Valdés, Damián Pérez. 2007. Maestros del Web. ¿Qué es Javascript? [En línea] 3 de julio de 2007. [Citado el: 2 de noviembre de 2016.] <http://www.maestrosdelweb.com/que-es-javascript>.

Villena, Antonio Moreno y. 2009. AEEMT. [En línea] 2009. [Citado el: 10 de octubre de 2016.] http://www.aeemt.com/contenidos_socios/Informatica/Informac_Informat_Tecnolog/AMV_AGI_AEEMT_HT%20ML_Historia.pdf.

Visual Paradigm. 2014. Visual Paradigm. *Visual Paradigm*. [En línea] Paradigm, 2014. [Citado el: 9 de noviembre de 2016.] <https://www.visual-paradigm.com/>.

Vladia, Ruvio. 2017. *Cuba.cu*. Aumenta el acceso de los cubanos a Internet. [En línea] 28 de enero de 2017. [Citado el: 1 de febrero de 2017.] <http://www.cuba.cu/otras/2017-01-28/aumenta-el-acceso-de-los-cubanos-a-Internet-/35049>

Wordpress. 2013. Características de Drupal. *Wordpress*. [En línea] 2013. [Citado el: 9 de noviembre de 2016.] https://codex.Wordpress.org/es:Caracter%C3%ADsticas_de_Wordpress.

BIBLIOGRAFÍA CONSULTADA

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

Acunetix. 2016. *acunetix*. [En línea] Acunetix, 2016. [Citado el: 9 de noviembre de 2016.]

Android Studio. 2011. *Android Studio*. Conoce Android Studio [En línea] 2011. [Citado el: 15 de noviembre de 2016.] <https://developer.android.com/studio/intro/index.html?hl=es-419>.

Alegsa, Leandro. 2016. Definición de Offline o fuera de línea (informática). [En línea] 27 de junio del 2016. [Citado el: 12 de enero de 2017] <http://www.alegsa.com.ar/Dic/offline.php>

Álvarez, Miguel Ángel. 2011. *DesarrolloWeb*. [En línea] 2011. [Citado el: 2 de noviembre de 2016.] <http://DesarrolloWeb.com>.

esepe studio. 2005. *esepestudio*. ¿Qué es MySQL? [En línea] martes, 16 de agosto de 2005. [Citado el: 1 de diciembre de 2016.] <http://www.esepestudio.com/noticias/que-es-mysql>.

ETECSA. 2017. Acceso público a Internet. *etecsa.cu*. [En línea] 2016. [Citado el: 1 de febrero de 2017.] http://www.etecsa.cu/Internet_conectividad/Internet/.

Estevan, Gomez, Ricardo. 2016. *espe.edu.ec*. Arquitectura de Android [En línea] 31 de mayo de 2016. [Citado el: 27 de marzo de 2017.] <https://sites.google.com/a/espe.edu.ec/programacion-ii/android/a-1---arquitectura-de-android>.

Fawkes, Guy. 2017. *vpnmentor.com*. Tendencias de Internet, estadísticas y datos en los Estados Unidos y el mundo para 2017 [En línea] 1 de enero de 2017. [Citado el: 24 de abril de 2017.] <https://es.vpnmentor.com/blog/page/2/>.

Foundation, Apache Software. 2011. Apache. *Apache JMeter*. [En línea] 09 de mayo de 2011. [Citado el: 9 de noviembre de 2016.] <http://jmeter.apache.org/>.

Gama y autores. 2003. *Reutilización de software*. Patrones de diseño. <http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf>

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. *neging.org*. [En línea] 2011. [Citado el: 8 de noviembre de 2016.] <https://www.nginx.com/>.

Ite. 2017. *Educacion.es*. Navegación offline. [En línea] 2017. [Citado el: 27 de febrero de 2017.] http://www.ite.educacion.es/formacion/materiales/157/cd/m2_1_navegacion/navegacin_offline.html.

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

jsx. 2016. *Mozilla*. Java script. [En línea] 2016. [Citado el: 2 de noviembre de 2016.] <https://developer.mozilla.org/es/docs/Web/JavaScript>.

Kauler, Leto. 2015. HTTrack website copier. [En línea] 2015. [Citado el: 27 de octubre de 2016.] <https://www.httrack.com>.

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

Letelier, Patricio. 2005. *Metodologías Ágiles y XP*. Valencia: Departamento de Sistemas Informáticos y Computación, 2005.

LibrosWeb,2006. *¿Qué es CSS?* s.l. : LibrosWeb.es, 2006. http://librosweb.es/libro/css/capitulo_1.html.

Ilián García, Raúl. *Diseño e implementación de un juego Android para soporte a la enseñanza de frameworks OO y patrones de diseño*. 2013.

Luca, Juan Ignacio de Tena. 2004. *Manual Imprescindible de PHP5*. Madrid: GRUPO ANAYA, S.A., 2004.

Marin, Marisela. 2012. *Búsqueda de información*. [En línea] 25 de junio del 2012. [Citado el: 17 de enero de 2017.] <http://pasoapasoatusalud.blogspot.com/>

Miani, Antonio Aliaga y Marco. 2011. I.E.Sanvicente. *PosgreSQL*. [En línea] 2011. [Citado el: 8 de noviembre de 2016.] <http://www.iessanvicente.com/colaboraciones/postgreSQL.pdf>.

Montiel, Daniel Ponsoda. 2008. *Introducción a SQLite*. s.l.: Creative Commons. 2008.

- Moreno, Manuel. 2017.** *Trecebits*. Mapa mundial del acceso a Internet. [En línea] 26 de enero de 2017. [Citado el: 31 de enero de 2017.] <http://www.trecebits.com/2017/01/26/mapa-mundial-del-acceso-a-Internet/>
- Muñoz de Frutos, 2016.** *computerhoy.com*. ¿Qué es APK? [En línea] 8 de octubre de 2016. [Citado el: 10 de febrero de 2017.] <http://computerhoy.com/noticias/moviles/que-es-apk-52222>
- Naramore, Elizabeth. 2005.** *Beginning PHP5, Apache, and MySQL Web Development*. Canadá: Wiley Publishing, 2005.
- Narom, José Luis. 2013.** *Narom. Cuántas paginas web hay en Internet 2013* [En línea] 2013. [Citado el: 31 de febrero de 2017.] <http://narom.com>
- NetBeans Community. 2000.** NetBeans Community. [En línea] 2000. [Citado el: 2 de noviembre de 2016.] <http://netbeans.org>.
- OMG. 2015.** *OMG. Introduction to OMG's*. [En línea] 24 de junio de 2015. [Citado el: 2 de noviembre de 2016.] http://www.omg.org/gettingstarted/what_is_uml.htm.
- phpmyadmin. 2003.** *phpmyadmin*. [En línea] 2003 a 2016. [Citado el: 9 de noviembre de 2016.] <https://www.phpmyadmin.net>.
- Peña. 2015.** *Paréntesis.com*. ¿Para qué sirve? [En línea] 8 de abril de 2015. [Citado el: 27 de febrero de 2017.] http://parentesis.com/tutoriales/Que_es_un_APK_y_como_se_instala.
- Popkin Software and Systems. 2007.** *tldp*. [En línea] 19 de 6 de 2007. [Citado el: 2 de noviembre de 2016.] <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.
- Rafael, Martínez. 2010.** *postgresql*. Sobre PosgreSQL. [En línea] 2 de octubre del 2010. [Citado el: 1 de diciembre de 2016.] http://www.postgresql.org.es/sobre_postgresql.
- Ramon, Jose. 2010.** *forobeta*. Simply Static - nuevo plugin, todos tus archivos "estáticos". [En línea] 14 de febrero del 2010. [Citado el: 27 de octubre de 2016.] <http://forobeta.com/plugins-wp/441107-simply-static-nuevo-plugin-todos-tus-archivos-estaticos.html>.

Redes sociales. 2016. *Franbarquilla*. ESTUDIO SOBRE EL ESTADO DE INTERNET Y LAS REDES SOCIALES EN EL 2016 [En línea] 2 de febrero de 2016. [Citado el: 31 de febrero de 2017.] <http://franbarquilla.com/estudio-sobre-el-estado-de-Internet-y-las-redes-sociales-en-2016/>.

Rodríguez Sánchez, Tamara. *Metodología de desarrollo para la Actividad productiva de la UCI*. 2014.

Robertson, James. 2008. Step Two Designs Pty Ltd. [En línea] 2008. [Citado el: 9 de noviembre de 2016.] http://www.steptwo.com.au/papers/kmc_what.

Rosa Donoso. 2016. *Androidpit*. ¿Qué es APK? [En línea] julio 2016. [Citado el: 1 de diciembre de 2016.] <http://www.androidpit.es/android-para-principiantes-apk>.

Stafford, Leon. 2015. *Wordpress*. WP Static HTML Output. [En línea] 2015. [Citado el: 27 de octubre de 2016.] <https://Wordpress.org/search/site/HTML%20Export>.

Sommariva, Andrés. 2014. *Pruebas Unitarias, Parte 1: Introducción y utilización de objetos simulados (Mock)*. [En línea] [Citado el: 26 de Abril de 2016.] [53] <http://www.microgestion.com/index.php/mg-developers/articulos/74-unit-test-part1-mock>.

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

Soto, Karen C. Zerquera. 2012. *revista.jovenclub.cu*. HTTrack WebSite Copier 3.44-1 (WebHTTrack). [En línea] 14 de marzo del 2012. [Citado el: 27 de octubre de 2016.] <http://revista.jovenclub.cu/htrack-website-copier-3-44-1-webhtrack/>.

The Eclipse Foundation. 2016. *Eclipse. About the Eclipse Foundation*. [En línea] 2016. [Citado el: 2 de noviembre de 2016.] <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: 9 de noviembre de 2016.]
<http://www.slideshare.net/StalinTuza/ingenieriasoftwareopenup#btnPrevious>.

UIT. 2015. Unión Internacional de Telecomunicaciones. *Measuring the Information Society Report*. [En línea] 2015. [Citado el: 12 de enero de 2017.]
https://www.itu.int/en/ITU-D/Statistics/Documents/publications/mis2014/MIS2014_without_Annex_4.pdf.

unicero. 2015. Unicero.com. ¿Qué es un archivo apk y como se instala?. . [En línea] 2015. [Citado el: 12 de enero de 2017.] <https://www.unocero.com/2015/09/03/que-es-un-archivo-apk-y-como-se-instala/>

Valdés, Damián Pérez. 2007. Maestros del Web. ¿*Qué es Javascript?* [En línea] 3 de julio de 2007. [Citado el: 2 de noviembre de 2016.] <http://www.maestrosdelweb.com/que-es-javascript>.

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

Vásquez, Eveling Giselles Cruz. 2013. Slideshare. *Slideshare*. [En línea] LinkedIn Corporation, 30 de noviembre de 2013. [Citado el: 9 de noviembre de 2016.]
<http://es.slideshare.net/EvelingGiselleCruzVs/metodologia-monografia>.

Villena, Antonio Moreno y. 2009. AEEMT. [En línea] 2009. [Citado el: 10 de octubre de 2016.]
http://www.aeemt.com/contenidos_socios/Informatica/Informac_Informat_Tecnolog/AMV_AGI_AEEMT_HT%20ML_Historia.pdf.

Visual Paradigm. 2014. Visual Paradigm. *Visual Paradigm*. [En línea] Paradigm, 2014. [Citado el: 9 de noviembre de 2016.] <https://www.visual-paradigm.com/>.

Wordpress. 2013. Características de Drupal. *Wordpress*. [En línea] 2013. [Citado el: 9 de noviembre de 2016.] https://codex.Wordpress.org/es:Caracter%C3%ADsticas_de_Wordpress.

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.

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

EXE: es una extensión que se refiere a un archivo ejecutable de código reubicable.

DMG: es muy sencillo porque este formato es el que usan, precisamente, los sistemas operativos de Apple para imágenes de disco.

Sitio de E-Commerce: es una tienda virtual en donde vendes productos de cualquier tipo.

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 Wordpress como unas funciones que permiten la comunicación entre los distintos plugins del sistema.

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

Plugins: 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 en Wordpress representa la interfaz de programación de aplicaciones de Wordpress. Se puede separar en múltiples *API* secciones / temas. Cada uno cubre las funciones involucradas y el uso de un conjunto dado de funcionalidad. Juntos forman lo que se llamar la *API* de Wordpress, que es el / tema / interfaz de complemento plug-in creado por la totalidad del proyecto de Wordpress.

GNU: la 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.

Ciente Enriquecido: consiste en proporcionar una interfaz gráfica, escrita con una sintaxis basada en XML, que proporciona funcionalidades similares a las del cliente pesado (arrastrar y soltar, pestañas, ventanas múltiples, menús desplegables).

Berkeley Software Distribution: La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution), un tipo del sistema operativo Unix-like. Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License.

Ciente-liviano: una aplicación a la que se puede acceder por una interfaz Web (en HTML), que se puede visualizar con un navegador Web en donde toda la lógica comercial se realiza en el lado del servidor.

GPL: es la licencia más ampliamente usada 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: proceso que posee 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.

Hardware: Corresponde a todas las partes físicas y tangibles de una computadora, sus componentes eléctricos, electromecánicos y mecánicos; sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente al soporte lógico e intangible que es llamado software.

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.

SSL: Secure Sockets Layer en español ``capa de puertos seguros`` son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

pg_upgrade: es una utilidad para actualizar una versión anterior de PostgreSQL sin la necesidad de recargar todos los datos. No todas las transiciones de versiones de Postgres se pueden manejar de esta manera. Verifique las notas de la versión para saber si hay detalles en su instalación.

IntelliJ IDEA: es un Ambiente de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (Anteriormente conocido como **IntelliJ**) y está disponible en dos ediciones: community edition y edición comercial.

Gradle: es básicamente una herramienta para automatizar el proceso de construcción de nuestro proyecto. Está basado en *Groovy* y el cual tiene una sintaxis muy similar a la de java.

Groovy: es un lenguaje de programación orientados a objetos.

GitHub: es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Framework: es un entorno o ambiente de trabajo para desarrollo; dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo de aplicaciones como el soporte de programa, bibliotecas, plantillas y más.

Google Cloud Platform: es, básicamente, una plataforma que ha reunido todas las aplicaciones de desarrollo web que Google estaba ofreciendo por separado, aumentando de esta forma su nivel de competitividad. Es utilizada para crear soluciones a través de la tecnología almacenadas en la nube.

App Engine: es un servicio de alojamiento web que presta Google de forma gratuita hasta determinadas cuotas. Este servicio permite ejecutar aplicaciones sobre la infraestructura de Google. Si no se cuenta con un dominio propio, Google proporciona uno con la siguiente estructura, `midominio.appspot.com`.

Google Cloud Messaging: es la mensajería en la nube de Google la cual se encarga de enviar mensajes a dispositivos Android.

W3C: Este validador verifica la validez de encajecimiento de documentos de Tejido en HTML, XHTML, SMIL, MathML, entre otros.

ANEXOS

Anexo N° 1. Entrevista

Tabla 5. 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 Wordpress?
3	¿Qué es un Tipo de Contenido y como está compuesto?

Anexo No 2. Diagramas de Clases del Diseño

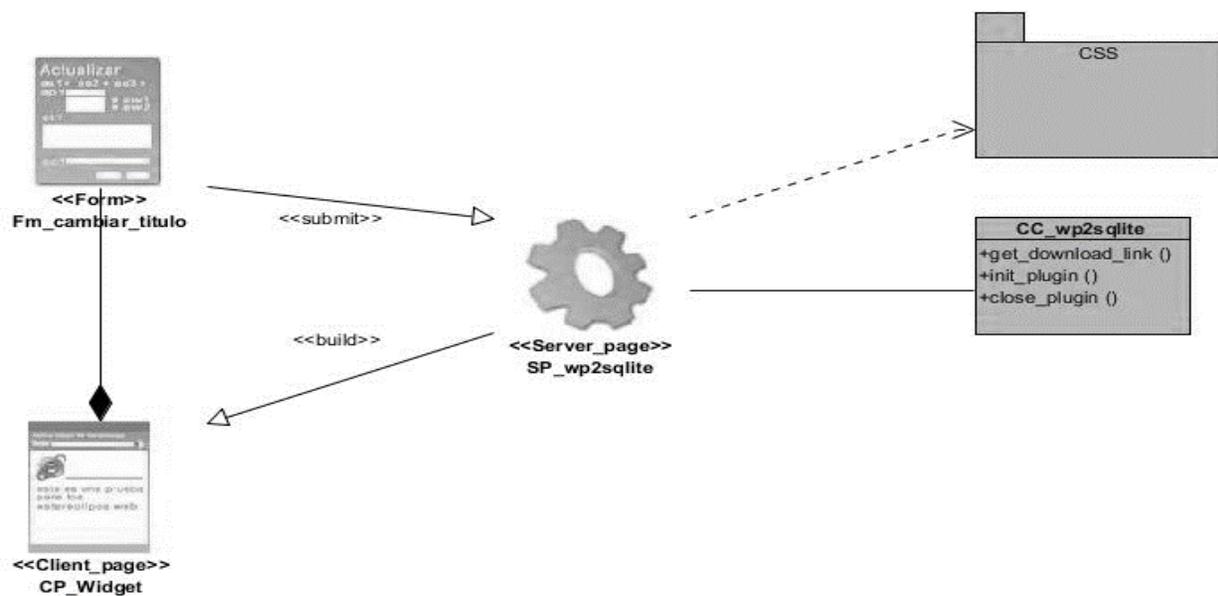


Figura 10. Diagrama de clases de diseño. Seleccionar Tipo de Contenido.

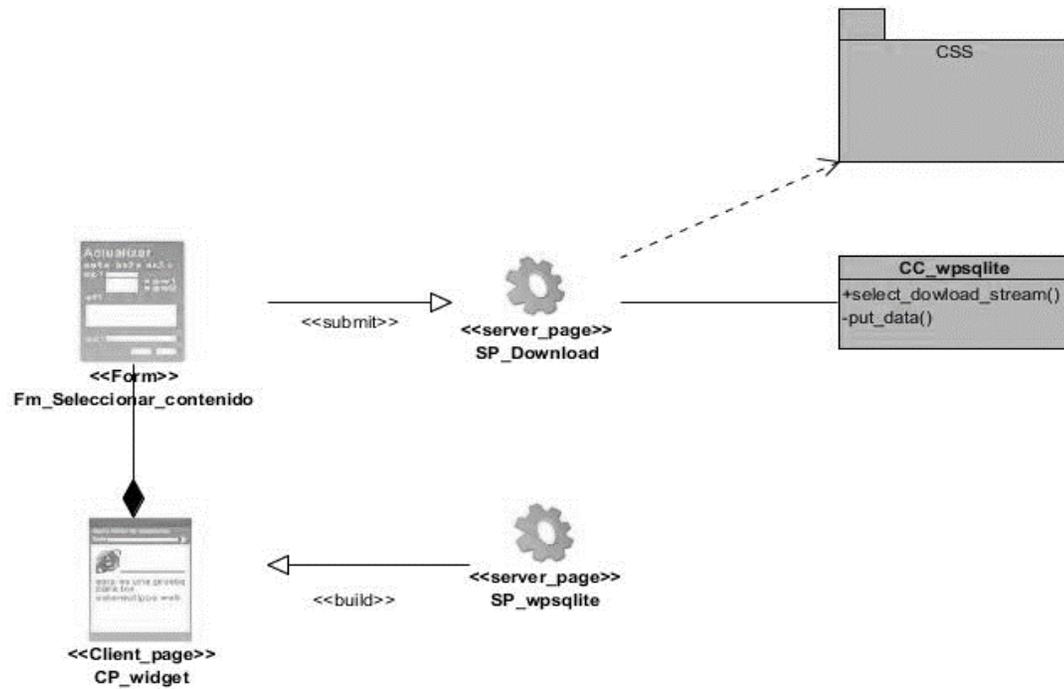


Figura 11. Diagrama de clases de diseño. Seleccionar Campos por Tipos de Contenido.

Anexo N°. 3. Pantallas del plugin.

Plugin WP2SQLite:

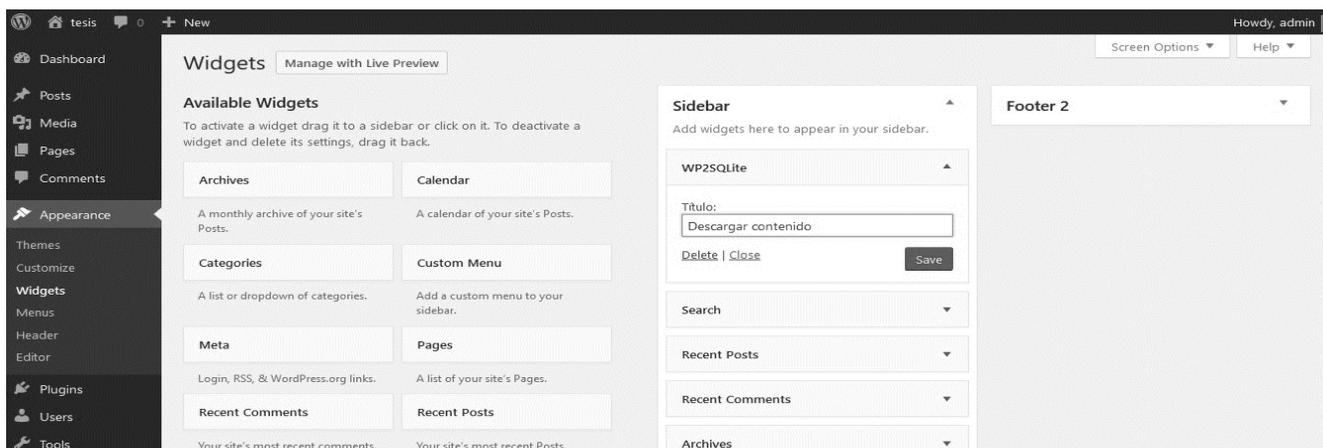


Figura 12. Modificar título de contenido a descargar.

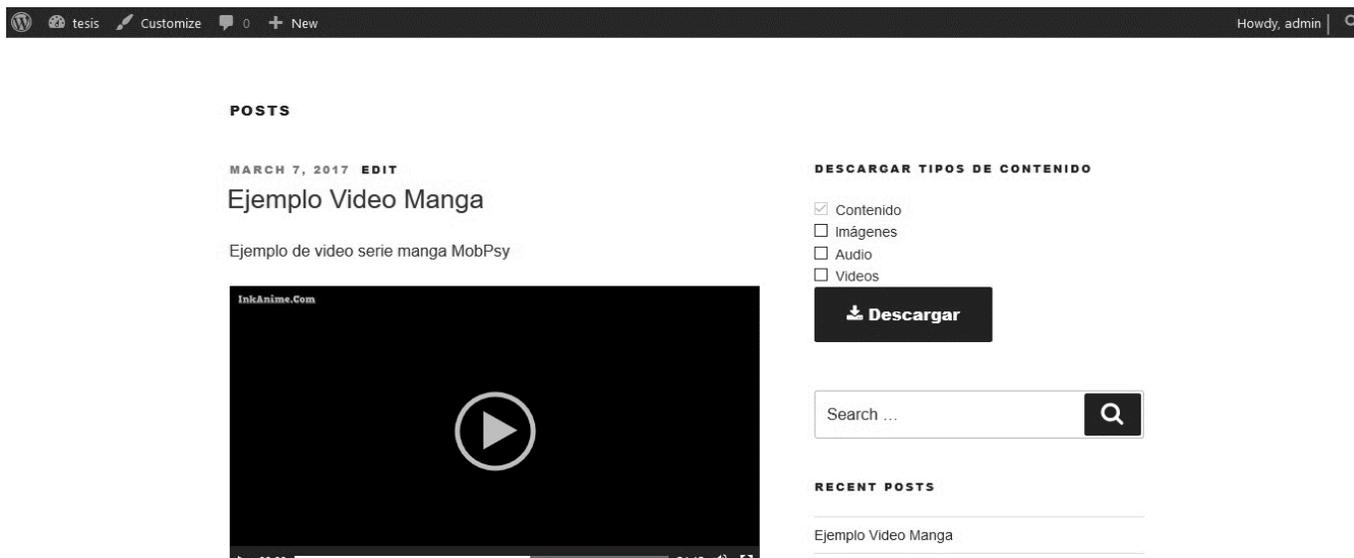


Figura 13. Seleccionar Campos del Tipo de Contenido a exportar.

Aplicación Android Wordpress contigo:

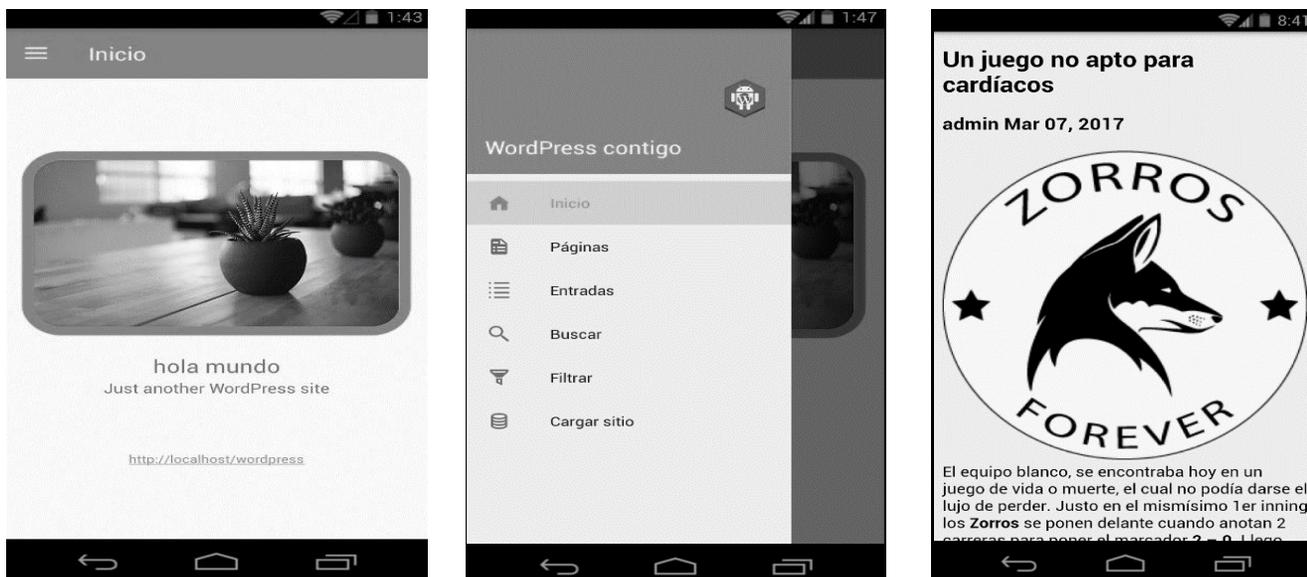


figura 14. Aplicación Android Wordpress contigo.