FACULTAD 1

Módulo para la detección de contenido duplicado en portales web

Trabajo de diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor:

Yoel Joaquin Silvente Lao

Tutores:

MSc. Delly Lien González Hernández

MSc. Hubert Viltres Sala

La Habana, junio de 2015

Declaración de autoría

Declaro ser autor de la presente tesis y ofrezco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma. Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

_____

Firma del autor
Yoel Joaquin Silvente Lao

_____

Firma del tutor
MSc. Delly Lien González Hernández

_____

Firma del tutor
MSc. Hubert Viltres Sala

Resumen

El presente trabajo consiste en el desarrollo de un módulo para la detección de contenido duplicado en portales implementados en Drupal 7. Se estudiaron varias herramientas existentes que detectan contenido duplicado y los módulos utilizados en Drupal para facilitar el posicionamiento. Se analizaron varios algoritmos de los reportados en la bibliografía y se seleccionaron la similitud coseno y el similar text porque combinados, disminuyen el tiempo de respuesta y se reducen las limitaciones individuales que presentan. Para la implementación se utilizó Drupal en su versión 7, y como guía del proceso la metodología de desarrollo de software OpenUp. Se emplearon como lenguajes de programación PHP 5, HTML 5 y CSS 3. La solución cuenta con PostgreSQL en su versión 9.1 y MySQL versión 5 como gestores de bases de datos. El modelado se realizó con la herramienta Visual Paradigm 8, el lenguaje de modelado fue UML y en las pruebas de seguridad se empleó la herramienta Websecurify. El módulo permite realizar una búsqueda para detectar la existencia de contenido duplicado sobre la base de los algoritmos seleccionados, comprueba la similitud entre los contenidos de las páginas del portal y ofrece un reporte con los nodos que presentan duplicado en caso de encontrarse. Esta solución brinda a los *webmasters* una herramienta que permite la detección de contenido duplicado en los portales para facilitar posicionamiento y optimización de la información. Las pruebas de software arrojaron que el módulo es funcional, seguro y se integra de forma adecuada a Drupal 7.

Palabras clave: contenido duplicado, *Search Engine Optimization*, detección, módulo, Drupal.

Índice

Introducción

Internet, con el devenir de los años, ha sufrido avances significativos. El medio que comenzó como un proyecto de propósitos más específicos, actualmente dispone de millones de usuarios. Según Internet World Stats (2014) existen aproximadamente en África 4 millones 514 mil 400 usuarios de internet, 114 millones 304 mil en Asia, 105 millones 96 mil 93 en Europa, 3 millones 284 mil 800 en el medio este, 108 millones 96 mil 800 en Norte América, 18 millones 68 mil 919 en Latino América y el Caribe y 7 millones 620 mil 480 en Australia y Oceanía para un total de 360 millones 985 mil 492 usuarios y la cifra de datos se aproxima a los 2802 millones 478 mil 934. Cada año que transcurre, mayor es el número de instituciones, empresas y usuarios que lo utilizan con el fin de publicar información.

De acuerdo con Royal Pingdom (2013) existen 634 millones de portales web y durante el transcurso de un año son publicados 51 millones de nuevos portales web en la red. El rápido crecimiento en la creación y publicación de portales web en internet, el auge de las herramientas de digitalización y la utilización de las redes sociales, han permitido que una mayor cantidad de información digital disponible sea accesible por los usuarios en internet. Esta tendencia de utilizar internet como una fuente de intercambio, publicación y obtención de conocimientos ha causado que en ocasiones se genere información repetida. La mala manipulación en la forma de publicar la información en un portal web por parte de los *webmasters* puede producir contenido duplicado.

Según plantean De Teresa (2014), Florido (2014) y Google (2014) el contenido duplicado se define como fragmentos de contenido en varios dominios que coinciden parcial o totalmente dentro o fuera de un portal web. Cuando un motor de búsqueda indexa varias páginas idénticas, se ve obligado a seleccionar de todas ellas cuál es la mejor opción para proporcionarle al usuario una buena experiencia en la búsqueda de información. La tendencia a publicar la información en la web y los errores cometidos por parte de los *webmasters* pueden traer consigo la aparición del contenido web duplicado que puede producirse dentro de páginas pertenecientes a un mismo portal web o en diferentes portales web donde se encuentra la misma información.

Los motores de búsqueda identifican el contenido duplicado a través de metadatos específicos como son: títulos, descripciones, encabezados y URL. La existencia de contenido web duplicado en un portal provoca

que se afecte la optimización para motores de búsqueda conocido como SEO, del inglés *Search Engine Optimization*. Con esto se afecta la posición del indexado realizado por los motores de búsqueda, pues puede producir error de presentación mostrándose una página duplicada no deseada, perdiendo enlaces a la página o una atribución de autoría equivocada. La duplicación de contenido constituye uno de los problemas del SEO más extendidos y contradictoriamente uno de los menos tratados. Generalmente cuando se habla de optimizar la página para los buscadores se piensa en conseguir enlaces pero nunca en eliminar el contenido duplicado (De Teresa, 2014; Enache, 2014; Florido, 2014; AdaptPartners.com, 2015; Businessonline.com, 2015).

El tal sentido, el desarrollo alcanzado por internet ha posibilitado que se genere un gran cúmulo de información en la red. En ocasiones esta información se duplica y los administradores de los portales web no perciben que pierden posicionamiento en los principales motores de búsqueda o simplemente malgastan espacio guardando una copia de un contenido que ya se encontraba disponible.

La demanda de portales web en general en el mundo, y en particular en Cuba, es creciente. Este hecho influye directamente en que los *webmasters* necesiten mantener un buen SEO de sus portales web y disminuir el espacio que ocupa el contenido duplicado para asegurar que el contenido mostrado en ellos sea accesible a un mayor número de usuarios y la capacidad de almacenamiento sea la óptima.

En la actualidad se conocen varias herramientas para identificar contenido duplicado en los portales web y que recomiendan acciones para tratarlo o eliminarlo, no obstante, no siempre son eficaces pues solo realizan la búsqueda de duplicado que se encuentren en diferentes dominios y no la búsqueda de duplicación interna en un mismo portal web. Asimismo realizan la búsqueda por un número determinado de caracteres, restringiendo las opciones de búsqueda y presentan limitaciones de licencia privativa. Por otra parte los sistemas de gestión de contenido o *Content Management System* (*CMS*) como Drupal, constituyen una forma fácil de realizar un portal web pero son propensos a la ocurrencia de contenido duplicado. Drupal, en tal sentido, presenta un módulo que es capaz de eliminar el contenido duplicado por URL (Global Redirect), pero solo tiene en cuenta el duplicado por URL dejando a un lado otros metadatos importantes como el título y el cuerpo de la página.

Por lo anteriormente planteado se define como **problema de investigación**: ¿Cómo detectar contenido duplicado en portales web desarrollados en Drupal 7 para contribuir a la optimización de la información disponible?

Se enmarca el **objeto de estudio** en el proceso de detección de contenido duplicado en portales web.

Se plantea como **objetivo general**: Desarrollar un módulo que permita detectar contenido duplicado en portales web desarrollados en Drupal 7 para contribuir a la optimización de la información disponible.

Por consiguiente los **objetivos específicos** son los siguientes:

1. Construir los referentes teóricos fundamentales que sustentan la investigación relacionados con el desarrollo de herramientas para la detección de contenido duplicado en portales web.
2. Diseñar las funcionalidades del Módulo de detección de contenido duplicado en portales web.
3. Implementar las funcionalidades del Módulo de detección de contenido duplicado en portales web.
4. Validar las funcionalidades del Módulo de detección de contenido duplicado en portales web.

Se selecciona como **campo de acción:** los portales web desarrollados en Drupal 7.

La **idea a defender** de la investigación radica en que el módulo para detectar contenido duplicado en portales web desarrollados en Drupal 7 influye directamente en la optimización de la información disponible.

Tareas de investigación:

1. Realización de un estudio sobre las tendencias en la detección de contenido duplicado.
2. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
3. Selección de la metodología de desarrollo.
4. Definición de los requisitos funcionales y no funcionales de la propuesta de solución.

5. Implementación de la propuesta de solución.
6. Documentación de las pruebas de funcionalidad, seguridad e integración que validan el módulo propuesto.

Los métodos científicos que se utilizan en la presente investigación son:

Métodos teóricos:

Analítico-Sintético: para el estudio de las fuentes bibliográficas y conceptos existentes sobre las herramientas de detección de contenido duplicado, con el objetivo de poder demostrar la necesidad de la investigación.

Análisis Histórico-Lógico: utilizado para realizar un estudio de cómo se comportan las herramientas para la detección de contenido duplicado en portales web y posibilitar la creación del marco teórico.

Inductivo-Deductivo: para el estudio de las principales iniciativas de detección de duplicado y los algoritmos utilizados para lograrlo con el objetivo de determinar cuáles son las alternativas viables a incorporar en la presente investigación.

Modelación: para la representación de la solución propuesta a través de la modelación de diagramas que permiten modelar el proceso a seguir para desarrollar la solución creando una abstracción con el objetivo de explicar la realidad.

Métodos empíricos:

Observación: utilizado para obtener información sobre los sistemas de detección de duplicado existentes. Observar los metadatos que utilizan, los algoritmos utilizados y el tipo de contenido duplicado que identifican.

Análisis documental: se empleó para la consulta de la literatura especializada en las temáticas afines a la investigación.

El presente documento se estructura en tres capítulos, además de las secciones de Introducción, Conclusiones, Recomendaciones, Bibliografía y Anexos.

Capítulo 1: Detección de contenido duplicado en portales web

Se realiza un análisis de los principales conceptos asociados a la detección de contenido duplicado en los portales web que son necesarios para la comprensión de la presente investigación. Se brinda una descripción sobre la existencia de duplicado en los portales web. Se presenta un estudio sobre los principales algoritmos que se utilizan para la detección del duplicado, las herramientas existentes para solucionar este problema y las afectaciones del duplicado web al SEO. Además se realiza un análisis de las herramientas, tecnologías y metodología para el desarrollo de software.

Capítulo 2: Análisis y diseño del módulo de detección de contenido duplicado en portales web

Se describe la propuesta de solución. Se especifican los requisitos funcionales y no funcionales. Se realiza el análisis y diseño a través de la transformación de los requisitos en casos de uso del sistema. Se muestran los diferentes diagramas de clases del diseño, diagramas de secuencia y de casos de uso.

Capítulo 3: Implementación y prueba del módulo de detección de contenido duplicado en portales web

Se realiza la implementación del módulo y se especifican las pruebas realizadas al mismo para comprobar que cumple con los requisitos planteados y asegurar la calidad del software, para ello se utilizan: diagramas de diseño, de despliegue, de componentes, así como los diferentes casos de prueba y resultados de los mismos.

Capítulo 1: Detección de contenido duplicado en portales web

La generación de contenido duplicado en los portales web proviene del aumento ascendente del uso de internet y como consecuente el incremento de la información en formato digital. En este capítulo se definirán conceptos asociados a la detección de contenido duplicado en portales web, algoritmos utilizados para su detección, herramientas existentes que realizan su detección, las afectaciones del duplicado web al SEO, las tecnologías y metodología de desarrollo de software empleadas.

1.1 El contenido duplicado en portales web

Según varios autores (De Teresa, 2014; Florido, 2014; Google, 2014) el contenido duplicado se define como fragmentos de contenido en varios dominios que coinciden parcial o totalmente dentro o fuera de un portal web.

Origen de la detección de duplicado en portales web

El creciente uso de internet y la tendencia de digitalizar la información para que esté disponible en la red hace crecer el cúmulo de información en internet constantemente. Este crecimiento trae implícito que se produzca el duplicado de la información web, pues la información a digitalizar puede encontrarse ya a disposición en la red o por error se pueden crear páginas duplicadas de un mismo contenido. Como principales causas de la ocurrencia de duplicado se pueden encontrar (Santiago, 2013; Carabaño, 2014; De Teresa, 2014):

Orígenes del contenido duplicado dentro de la página web

- Dominio preferido: se puede acceder a una página mediante las versiones con o sin www de la URL.
- Páginas https: si la página web usa encriptación SSL, puede acabar con una copia exacta de la página web en la versión segura.
- IDs de sesión: algunas páginas web manejan las sesiones de usuario introduciendo un código al final de la URL de cada página. Estos parámetros, diferentes para cada sesión de usuario, hacen que el buscador crea que se trata de páginas separadas, aunque en realidad es la misma.

- Contenido dinámico: existen páginas web que asignan parámetros a las URL para controlar el contenido que muestran las páginas al usuario. De la misma manera que ocurre con los IDs de sesión, los buscadores interpretan muchas de estas páginas como copias.
- Archivos: el mismo contenido puede aparecer en páginas diferentes, como ocurre en los archivos de categorías y etiquetas.
- Paginación: cualquier página web que utilice paginación puede tener este problema, especialmente si las diferentes páginas comparten el mismo título y la misma descripción.

Orígenes del contenido duplicado fuera de la página web

- Sindicación: consiste en enviar los contenidos a otras páginas web para atraer tráfico, como por ejemplo mediante RSS. El problema puede surgir cuando estas páginas web publican una copia completa de tu contenido, en lugar de un fragmento.
- Localización: presentación del mismo contenido en diferentes dominios.
- Scrapers: son personas que mediante un software robot copian una página web para publicarla en otro dominio.
- Plagios: copiar un texto de una página web y publicarlo en otro como propio.

La ocurrencia de contenido duplicado afecta gravemente el SEO que es la actividad de optimizar páginas web para hacerlas más fáciles de buscar por los motores de búsqueda obteniendo posiciones más elevadas en los resultados de búsqueda (Enache, 2014). La ocurrencia de este problema afecta considerablemente a los portales a la hora del indexado por parte de los motores de búsqueda.

Como se ha podido observar diversos son los orígenes de la aparición de contenido duplicado en los portales web, factor que afecta el SEO. Importante es además, conocer las consecuencias que puede traer consigo la aparición de duplicado y las medidas que se pueden tomar para evitar su existencia. Conocimientos que nos permitirán mejorar el posicionamiento web (Santiago, 2013; Carabaño, 2014; De Teresa, 2014).

Consecuencias del contenido duplicado:

- Páginas incorrectas: tener diferentes páginas para un mismo contenido significa dejar en manos del buscador la elección de la página correcta.
- Peor visibilidad: como consecuencia de las páginas incorrectas, el buscador puede acabar mostrando una copia con menor peso que la página deseada, y por tanto, posicionarla peor de lo que estaría la versión correcta.
- Indexación deficiente: la indexación de las páginas puede verse afectada debido a que el buscador invierte su tiempo rastreando páginas duplicadas, en lugar de las páginas que importan.
- Desperdicio de enlaces: las páginas duplicadas pueden recibir enlaces y diluir la fuerza de los contenidos, ya que todos esos enlaces podrían estar sumando fuerzas en una única página.
- Atribución equivocada: el buscador puede decidir que el contenido es originario de un dominio que no es el tuyo.

Medidas para evitar el contenido web duplicado

- Etiqueta “rel=canonical”.
- Redirecciones 301.
- Desindexación.
- Gestionar los parámetros de URL.
- Unificar páginas o reescribir contenidos.

La cantidad de portales publicados es ascendente y muchas veces contienen la misma información repetida dentro de ellos mismos o copiada de otros portales. Debido al creciente problema de duplicación de contenido en la red y a las afectaciones que provoca su aparición son diversas las herramientas existentes para la detección de contenido duplicado en portales web. Ellas se encargan de realizar una búsqueda interna o externa de los contenidos para determinar si existe duplicidad parcial o total en los mismos.

1.2 Estudio de las herramientas existentes

A continuación se analizan algunas de las herramientas utilizadas para la detección de contenido web duplicado, las cuales se encuentran entre las recomendadas en portales sobre SEO y *marketing* digital (Valero, 2011; Carabaño, 2014; Clemente, 2014). A saber:

- Copyscape.
- Duplichecker.
- Google *webmasters tool.*
- Plagiarismchecker.
- Plagium.
- Screaming Frog

Copyscape

Fue lanzado en el 2004 por Indigo Stream Technologies, Ltd. Es un sistema de detección de plagio *online* que chequea la existencia de cualquier texto similar en la web y es utilizado para detectar casos donde el contenido ha sido copiado sin autorización de un portal a otro. Funciona utilizando un conjunto de algoritmos para identificar contenido copiado que ha sido modificado de su forma original y para ello utiliza el metadato URL o un texto específico. Posee una interfaz similar a la de Google, utiliza como proveedores de búsqueda a Google y Yahoo! y devuelve como resultado un listado con los portales que poseen una copia del contenido. Ofrece además la opción de un banner para agregar en el portal y advertir a los plagiadores que se está monitoreando el contenido. Sin embargo, presenta un plan gratuito que es limitado y permite solamente comparar el contenido de dos URL en concreto, si el contenido no se encuentra publicado no se puede comparar utilizando su versión gratuita.

Duplichecker

Herramienta gratuita para detectar el contenido duplicado en la web que recomienda para un rápido procesamiento en el modo *Advanced*, utilizar Mozilla Firefox. Entre sus opciones se encuentran:

- Realiza búsquedas en un buscador en específico (Google, Bing o Yahoo!).

- Realiza búsqueda a través de un texto que puede ser pegado directamente en una caja de texto o subiendo un archivo.
- Realiza la búsqueda en portales externos que contengan el mismo contenido o párrafos del contenido.
- El límite máximo de caracteres de búsqueda es de 1500 palabras por búsqueda.

Google *webmasters tools*

Es un conjunto de herramientas gratuitas de Google. Muestra todos los errores que encuentra al leer un portal para que seas capaz de resolverlo de forma rápida y sencilla. Las herramientas permiten:

- Analizar tu tráfico en la web.
- Determinar cuáles son las consultas más populares que se hacen en tu página.
- Optimizar la estructura.
- Obtener un informe de las diferentes palabras clave y frases utilizadas por el visitante al entrar en tu web.
- Medir la tasa de conversión.
- Control y seguimiento de los vínculos que apuntan a tu página.
- Ver con qué frecuencia enlaza la gente a tu página.
- Detectar software dañino, errores y todo lo que puede afectar a tu posicionamiento web.
- Eliminar o añadir direcciones URL.

Google *webmasters tools* se encuentra dividido en 4 bloques:

1) Configuración
    - La Orientación Geográfica.
    - Dominio preferido.
    - Enlaces de portales.
    - Parámetros de URL.
2) Estado

- Errores de rastreo.
- URL bloqueadas.
- Explorar como Google.
- Estado de indexación.

3) Tráfico

- Consultas de búsquedas.
- Enlaces externos al portal.
- Enlaces internos.

4) Optimización

- *Sitemaps*: para enviar el sitemp.xml de la web y brinda información del rastreo: fecha, incidencias URL enviadas y URL indexadas.
- Eliminación de URL: para eliminar una URL que el buscador haya indexado y se quiere eliminar.
- Mejoras de HTML: útil para detectar metadatos duplicados.

Plagiarismchecker

Herramienta para la detección de duplicado en la web. Creado por la Universidad de Maryland. Entre sus características podemos encontrar:

- Posee un límite de una búsqueda por día en su versión gratis.
- Permite revisar un documento o una página web.
- Capacidad de seleccionar el motor de búsqueda (Google o Yahoo!).
- Crear alertas.
- Solo busca hasta un máximo de 32 palabras, y puede no detectar todo el contenido duplicado.

Plagium

Herramienta para detectar el contenido duplicado en la web. Posee opciones de búsqueda como son:

- Ofrece dos opciones de búsqueda: *Quick Search* (Búsqueda rápida) y *Deep Search* (Búsqueda profunda).
- Selección del buscador a revisar (Google, Bing o Yahoo!).
- Permite buscar a partir de un texto o de una URL.
- Configuración de alertas de correo electrónico para notificar cuando el contenido es copiado.
- Permite personalizar la búsqueda en: la web, noticias o redes sociales.
- Realiza búsquedas en páginas en idioma: inglés, francés, portugués, alemán, italiano y español.
- Posee dos planes: uno gratuito y uno pago. Sin embargo si necesitas hacer diversas comprobaciones en una jornada necesitas contratar el servicio pago *Premium*.

Screaming Frog

Herramienta utilizada fundamentalmente para el SEO pero también permite detectar contenido duplicado; es capaz de ofrecer datos de un portal como son:

- Links rotos.
- Contenido duplicado.
- Enlaces externos.
- Número de etiquetas H1, H2.
- Información de imágenes.
- Errores de indexación (404, 500).
- Estado de las *meta-tags*.
- Estado de las etiquetas *title, description y keywords*.
- Análisis del archivo robots.txt.

Todas las herramientas analizadas presentan limitaciones; Copyscape, Plagiarismchecker y Plagium presentan limitación de licencia privativa pues solo permiten acceder a todas sus funcionalidades a través del pago de una licencia; Screaming Frog solo permite detectar contenido duplicado exacto sin analizar los que presenten similitud parcial además, en su mayoría solo enfocan la detección de duplicado en detectar

el contenido duplicado entre portales para detectar el contenido plagiado y no el duplicado interno en portales, factor que afecta al portal considerablemente en el SEO y además derrocha espacio de almacenamiento en contenido duplicado. Un elemento importante en el análisis de las herramientas existentes consiste en que ninguna de ellas está enfocada hacia la integración con Drupal, sistema de gestión de contenido utilizado por el Centro de Ideoinformática (CIDI) para el desarrollo de sus portales web.

Luego de este análisis se pudo observar que no existe ninguna herramienta capaz de detectar el contenido web duplicado dentro del mismo portal y que además permitiera su integración con Drupal. Por esta razón se decide realizar la herramienta que brinde solución a la detección de contenido web duplicado dentro de un mismo portal, para facilitarle a los clientes que utilicen los portales desarrollados por CIDI, la detección del contenido web duplicado y que permita a los *webmasters* poder corregir los nodos que presenten contenido duplicado para que los portales no presenten problemas de duplicidad. Para el desarrollo de la solución, luego del análisis de las herramientas homólogas se identificaron como características a incluir en la solución final:

- Capacidad de realizar una búsqueda de contenido web duplicado a través de metadatos específicos (URL, título, cuerpo de página).
- Interfaz amigable.

Las herramientas analizadas utilizan para la detección de duplicado algoritmos de similitud y distancia, que son representaciones matemáticas que posibilitan el cálculo de la similitud entre cadenas utilizando diferentes algoritmos para ello y teniendo en cuenta un umbral para determinar la similitud que presentan las cadenas analizadas (Benavides, 2014).

1.3 Algoritmos de detección de similitud

Los algoritmos de detección de similitud pueden utilizar funciones de similitud basadas en el emparejamiento de cadenas por caracteres, *tokens* o *fuzzy hashing*. Las funciones de similitud basadas en caracteres consideran cada cadena como una secuencia ininterrumpida de caracteres. Las funciones de similitud basadas en *tokens* consideran cada cadena como un conjunto de subcadenas separadas por caracteres especiales como es el caso de: espacios en blanco, puntos y comas. Ellas calculan la similitud

entre cada pareja de *tokens* mediante una función de similitud basada en caracteres. Por otra parte los que utilizan técnicas de *fuzzy hashing* obtienen una huella digital de la cadena de texto a través de funciones *hash* y realizan la comparación de las huellas.

A continuación se analizan algunas de las técnicas que se encuentran dentro de la categoría de funciones basadas en caracteres, *tokens* o *fuzzy hashing*.

1. Por caracter:
    - Distancia de Levenshtein, distancia de edición o distancia entre palabras.
    - Similitud Smith–Waterman.
    - Distancia de brecha afín.
    - Similitud de Jaro.
    - Similitud de q-grams.
2. Por *tokens*:
    - Similitud de Monge-Elkan.
    - Similitud coseno.
3. *Fuzzy hashing:*
    - Similar text.

Distancia de Levenshtein

La distancia de edición entre dos cadenas se basa en el conjunto mínimo de operaciones de edición necesarias para transformar una palabra **A** en una palabra **B** o viceversa. Las operaciones de edición tienen un costo unitario y las permitidas son: inserción, eliminación y sustitución. Entre más cercano a cero es el valor devuelto por la función más parecidas son las palabras.

Similitud Smith–Waterman

La similitud entre dos cadenas **A** y **B** es la máxima similitud entre una pareja (**A´**, **B´**) sobre todas las posibles, tal que **A′** es subcadena de **A** y **B′** es subcadena de **B**. Problema que es conocido como alineamiento local. El modelo define las mismas operaciones que la distancia de Levenshtein y además permite la omisión de caracteres al principio o al final de ambas cadenas (Smith, 1981).

Distancia de brecha afín

Ofrece una solución para identificar cadenas que han sido demasiado truncadas por el uso de abreviaturas o la omisión de *tokens*, penaliza la inserción o eliminación de n caracteres consecutivos (brecha) con un bajo costo mediante una función afín p (k)= g + h *(k-1), en donde g representa el costo de iniciar una brecha, h el costo de extenderla un caracter y k<g. La distancia de brecha afín no normalizada puede ser calculada con un orden de O (nm) (Gotoh, 1982).

Similitud de Jaro

Define una función de transposición entre dos caracteres como única operación de edición permitida. Los caracteres no necesitan ser adyacentes sino que pueden estar alejados a una distancia d que depende de la longitud de las cadenas. La similitud de Jaro puede ser calculada con un orden de complejidad de O(n). La distancia entre dos cadenas S1 y S2 se puede calcular mediante (Jaro, 1976):

d= 0 si m =0.

d= $(1/3)$*$(\frac{m}{|s1|}+\frac{m}{|s2|}+\frac{m-t}{m})$ en otro caso.

d=distancia de jaro.

m= número de caracteres que emparejan.

t= la mitad del número de transposiciones.

s1= número de caracteres de la palabra 1.

s2= número de caracteres de la palabra 2.

Cada caracter de S1 se compara con todos sus caracteres coincidentes en S2. El número de coincidencias dividido por 2 define el número de transposiciones. Dos caracteres de S1 y S2 respectivamente, se consideran que emparejan sólo si tienen el mismo valor definido por la fórmula:

$$\left|\frac{\max(|S1|,|S2|)}{2}\right|-1$$

Similitud de q-grams

Los q-grams constituyen subcadenas de longitud q. La función de este se basa en que cuando dos cadenas son similares tienen muchos q-grams en común. Los más comunes son los uni-grams (q=1), bi-grams (q=2), tri-grams (q=3). Es posible agregar *q* – 1 ocurrencias de un caracter especial (no definido en el alfabeto Σ original) al principio y final de ambas cadenas. Esto llevará a un puntaje de similitud mayor entre cadenas que compartan algún prefijo o sufijo, aunque presenten diferencias hacia el medio (Yancey, 2006).

Similitud de Monge-Elkan

Dadas dos cadenas **A** y **B**, sean α1, α2…αk y β1, β2…βl sus *tokens* respectivamente. Para cada *token* αi existe algún βj de máxima similitud. Entonces la similitud de Monge-Elkan entre **A** y **B** es la similitud máxima promedio entre una pareja (αi, βj). El algoritmo tiene un orden computacional O (nm) (Monge, 1996).

Similitud coseno

Dadas dos cadenas **A** y **B**, sean α1, α2…αk y β1, β2…βl sus *tokens* respectivamente, que pueden verse como dos vectores Va y Vb de k y l componentes. Se propone una función que mide la similitud entre **A** y **B** como el coseno del ángulo que forman sus respectivos vectores (Cohen, 1998).

$$coseno\ (a,b) = \frac{\sum_{i=1}^{N} a_i b_i}{\sqrt{\sum_{i=1}^{N} a_i^2}_{\ i} \sqrt{\sum_{i=1}^{N} b_i^2}}$$

Proporciona un valor igual a 1 si el ángulo comprendido es cero. Si los vectores son ortogonales el coseno se anula, y si apuntan en sentido contrario su valor es -1. El valor de esta métrica se encuentra en el intervalo [-1,1].

Según Cohen (2003) proponen una variante llamada SoftTF-IDF que tiene en cuenta parejas de *tokens* (αi, βj) cuya similitud es mayor que cierto umbral mediante una función de similitud basada en caracteres.

Similar text

Dadas dos cadenas de texto **A** y **B**, el algoritmo calcula una suma de control basados en el contexto desencadenada por tramos *hashes* para cada entrada. Posteriormente compara los *hashes* y devuelve el porciento de similitud que existe entre ellas.

Para el desarrollo de la solución de software de este trabajo se decidió utilizar una combinación de los algoritmos similar text y similitud coseno, en ese orden respectivamente, para optimizar la búsqueda de contenido duplicado. Se escogen porque son algoritmos potentes para la detección de documentos duplicados que se utilizan con frecuencia en la búsqueda y recuperación de información y para establecer métricas de semejanza entre textos en la minería de textos. Además se suelen emplear como un indicador de cohesión de clústeres de textos. Mediante la combinación de estos algoritmos se consigue disminuir el tiempo de respuesta y se reducen las limitaciones individuales que presentan.

En el transcurso de esta investigación se analizó que Drupal solo presenta un módulo que es el más cercano al tratamiento del contenido duplicado (Global Redirect), que solo tiene en cuenta el duplicado por URL, por lo que no posee ninguna solución que permita detectar el duplicado a través de otros campos importantes como son el título y cuerpo de una página. No obstante presenta módulos que adicionan funcionalidades que le permiten a los *webmasters* realizar tareas que faciliten el posicionamiento de sus portales web.

1.4 Módulos para mejorar el posicionamiento web disponibles en Drupal

Como se ha podido analizar con anterioridad la duplicación de contenido es un factor que afecta directamente el SEO. Por esta razón a continuación se describirán algunos de los módulos que facilitan el SEO en el *CMS* Drupal, sistema al que está dirigido el resultado de la investigación (Gil, 2012).

Módulo XML sitemap

Genera un mapa del portal en formato XML según el protocolo definido en sitemaps.org. Los buscadores utilizan este protocolo para mantener los resultados indexados actualizados. Además, algunos buscadores como Google, Bing o Yahoo! permiten el envío directo del mapa del portal para ser utilizado en la indexación de sus contenidos. El módulo se compone de un conjunto de módulos que permiten crear mapas muy completos:

- **XML sitemap**: módulo base, necesario para generar el mapa del portal XML.
- **XML sitemap custom**: permite añadir enlaces personalizados al mapa.
- **XML sitemap engines:** permite enviar el mapa a distintos motores de búsqueda.
- **XML sitemap internationalization**: mapas del portal en función del idioma.
- **XML sitemap menú**: añade elementos de menú al mapa del portal.
- **XML sitemap node**: añade los enlaces a nodos al mapa del portal.
- **XML sitemap taxonomy**: añade los enlaces a términos de taxonomía al mapa del portal.
- **XML sitemap user**: añade los enlaces a los perfiles de usuario al mapa del portal.

Módulo Search 404

Sustituye la página de error 404 por una operación de búsqueda con el resultado de buscar la URL no encontrada en el portal.

Módulo Global Redirect

Evita ofrecer al motor de búsqueda URL duplicadas sobre un mismo contenido. Genera en las páginas con la URL sin alias, una redirección 301 a la URL con alias, indicándole al buscador que la URL antigua ha cambiado y se establece un enlace permanente a la nueva.

Módulo Meta tags

Añade metadatos estructurados en cada página del portal. Incluye soporte para etiquetas Open Graph (a través del módulo **Open Graph meta tags**), que es el protocolo utilizado por Facebook para obtener información de la página al compartirla en la red social. Añade valores por defecto para las meta etiquetas de la página principal, los contenidos, términos de taxonomía y perfiles de usuarios. Además permite añadir etiquetas por defecto para elementos determinados, como, por ejemplo, para un tipo de contenido específico.

Módulo Page Title

Permite añadir patrones para generar automáticamente el título de cualquier página (<title></title>). Haciendo uso de comodines permite componer los títulos de las páginas para distintos elementos:

- Página principal.
- Páginas con paginado.
- Páginas de respuesta a comentario.
- Valores específicos según el tipo de contenido.
- Perfiles de usuario.
- Páginas de términos de taxonomía.

Módulo Pathauto

Este módulo depende del módulo Token. Permite generar alias de URL de forma automática. Los alias se generaran a partir de información del propio nodo (por ejemplo, a partir del título), el módulo realizara una conversión de caracteres para evitar que la URL tenga caracteres especiales, espacios, etcétera.

Módulo Custom breadcrumbs

Permite crear migas de pan personalizadas según el tipo de contenido.

Módulo SEO Checklist

Genera un *checklist* con módulos de Drupal recomendados y buenas prácticas para el posicionamiento SEO. El *checklist* se divide en grupos de elementos: títulos de páginas, rutas de URL, contenido, protección de *Spam,* etcétera.

Módulo SEO Compliance Checker

Comprueba, al crear o editar un contenido, si éste cumple con las reglas SEO definidas. SEO Compliance Checker se compone de tres módulos:

- **SEO Compliance Checker:** módulo base, que comprueba las reglas definidas al crear o editar un contenido. Este módulo no añade reglas, por lo que tendremos que instalar conjuntamente los otros módulos incluidos o módulos adicionales que hagan uso de su API.
- **Basic SEO Rules:** incluye reglas básicas de posicionamiento.
- **Keyword Rules:** incluye reglas para comprobar la correcta utilización de las palabras clave en el contenido.

Módulo SEO Watcher

Busca las palabras clave especificadas en algunos motores de búsqueda y devuelve un informe con la posición del portal con respecto a la búsqueda realizada. Permite programar una comprobación de *ranking* en los principales buscadores (Google, Yahoo! y Bing) que se ejecutará periódicamente, enviando un correo con el informe a los destinatarios indicados.

Módulos Content Optimizer y Content Analysis

Content Optimizer analiza el contenido de cualquier página del portal y devuelve un informe con recomendaciones para que el contenido cumpla con las buenas prácticas de SEO. El módulo requiere la API facilitada por Content Analysis.

El uso de estos módulos les permite a los *webmasters* disponer de herramientas capaces de facilitar las labores de SEO. Pues algunos de estos brindan recomendaciones sobre módulos necesarios para el SEO y buenas prácticas para el SEO, comprueban si los contenidos creados cumplen con reglas SEO definidas, evitan ofrecer a los motores de búsqueda URL duplicadas y brindan información a los buscadores que les ayuda a indexar el contenido del portal. A pesar de que cada uno por separado es de gran utilidad, actualmente no existe ningún módulo en Drupal 7 que sea capaz de realizar las comprobaciones necesarias que permitan detectar la ocurrencia del contenido web duplicado dentro de un portal web.

1.5 Tecnologías y metodología asociadas al desarrollo de la solución

Hoy día existe un conjunto de tecnologías potentes para el desarrollo de aplicaciones web como lo son los sistemas de gestión de contenido. Estos permiten crear y mantener un portal web con mayor facilidad pues con ellos se pueden realizar tareas en un menor espacio de tiempo.

CMS Drupal 7

Drupal es un sistema de administración de contenidos que posibilita crear, clasificar y publicar información en un portal web. Distribuido bajo la licencia de GNU *Public License* (GPL). Dispone de varias versiones de su núcleo siendo las ramas 6 y 7 de sus últimas liberaciones. Presenta una estructura modular que le permite mediante un amplio repositorio de módulos añadirle nuevas funcionalidades.

Presenta un gran catálogo de módulos y temas gráficos para un sin número de prestaciones gratuitos de pago. Entre sus beneficios podemos encontrar (Drupal.org, 2014):

- Código abierto: presenta una comunidad de más de 1 millón de usuarios y desarrolladores.
- Posee herramientas para organizar, estructurar, encontrar y reutilizar el contenido.
- Permite la categorización de contenidos.
- Creación automática de URL amigables.
- Opciones de creación de cuentas de usuarios y roles.
- Maneja gran número de tipos de contenido, como video, texto, *blogs*, *podcasts*, manejo de menú, estadísticas en tiempo real y control de revisiones opcional.
- Posee más de 16000 módulos disponibles.
- Multiplataforma.
- Internacionalización.
- Soporta varios gestores de bases de datos (SQLite, PostgreSQL, MySQL, MariaDB o equivalente).
- Gran variedad de temas.

Lenguaje *Hypertext Pre-processor* 5 (PHP)

Lenguaje de código abierto adecuado para el desarrollo web que puede ser incrustado en el HTML y permite la creación de páginas web dinámicas. Entre las características por las cuales se selecciona este lenguaje se encuentran:

- Drupal está implementado en PHP.
- Multiplataforma.
- Presenta soporte para bases de datos como: MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

Lenguaje *HyperText Markup Language* 5 (HTML)

Algunas de sus características son:

- Estructura del cuerpo: permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada una de las partes típicas de una página.

- Etiquetas para contenido específico: utiliza etiquetas específicas para cada tipo de contenido en particular, como audio, video, etcétera.
- Bases de datos locales: permite el uso de una base de datos local, con la que se puede trabajar en una página web por medio del cliente y a través de un API.
- Aplicaciones web *offline*: permite la creación de aplicaciones web que funcionen sin necesidad de estar conectados a internet.
- Geolocalización: permite localizar geográficamente las páginas web por medio de un API.

Lenguaje *Cascading Style Sheets* 3 (CSS)

Es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo utilizado para especificar el aspecto de una página web. Se basa en reglas que rigen el comportamiento del estilo de los elementos.

Novedades de CSS 3 (Campos, 2011):

- Bordes:
    - Colores múltiples de borde en un mismo lado.
    - Imágenes de borde.
    - Bordes redondeados.
- Fondos:
    - Fondos múltiples pueden ser añadidos al mismo elemento como capas.
    - Posicionamiento del fondo con mayor precisión.
    - Pueden ser redimensionados.
- Color:
    - Opacidad.

- Gradientes.
- Valores de color: HSL.

- Text:
  - Sombras.
  - Desbordamiento.
  - Ajuste de línea.
- Transformaciones:
  - Escalar.
  - Sesgar.
  - Mover.
  - Rotar en 2D o 3D.
- Transiciones:
  - Transición sencilla de estilos.
- Cajas:
  - Sombras.
  - Cajas redimensionables.
  - *Overflow* separado en vertical u horizontal.
  - Compensación entre contorno y borde.
  - Modelos para especificar altura y anchura.
- Contenido:
  - Los estilos pueden añadir contenido a los elementos.
- Opacidad:

- Los elementos pueden ser transparentes.

- Fuentes web:

    - Capacidad de añadir fuentes en vivo a los documentos mejorada.

Sistemas gestores de bases de datos (SGBD)

Los SGBD se definen como el conjunto de programas que administran y gestionan la información contenida en una base de datos (ALVAREZ, 2007).

PostgreSQL 9.1

Poderoso gestor de bases de datos de código abierto. Funciona sobre sistemas operativos como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. Entre sus características podemos encontrar (Postgresql.org, 2010):

- Soporte para distintos tipos de datos.
- Incluye herencia entre tablas.
- Copias de seguridad en caliente (*Online/hot backups*).
- Regionalización por columna.
- *Multi-Version Concurrency Control* (MVCC).
- Múltiples métodos de autenticación.
- Acceso encriptado vía SSL.
- Licencia BSD.

MySQL 5

Sistema de gestión de bases de datos ampliamente utilizado en aplicaciones web, como Drupal. Funciona sobre plataformas como: AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, Windows, etcétera. Permite (Mysql.org, 2015):

- Amplio subconjunto del lenguaje SQL.
- Disponibilidad en gran cantidad de plataformas y sistemas.

- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad y distribución geográfica.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Metodología OpenUP

Metodología ágil propuesta por un conjunto de empresas (IBM Corp., Telelogic AB., Armstrong Process Group, Inc., Number Six Software, Inc. & Xansa plc.). Forma parte del *framework* de modelo de proceso de Eclipse (*Eclipse Process Framework*). Aplica un enfoque iterativo e incremental dentro de su ciclo de vida. Solo genera la documentación fundamental por lo que no provee guías para aspectos como: equipos de desarrollo de gran tamaño, situaciones contractuales, aplicaciones críticas o de salud o guías específicas sobre tecnología. Como es una metodología incremental, su proceso puede ser personalizado según las necesidades que aparecen a lo largo del ciclo de vida del software. Una de sus mayores ventajas la constituye que puede ser utilizada para proyectos pequeños y que es capaz de generar versiones. La metodología OpenUP está basada en la metodología *Rational Unified Process* (RUP) y de esta mantiene características como son: diseño basado en la arquitectura, utilización de casos de uso y escenarios, manejo de riesgos y desarrollo incremental. En la misma pueden distinguirse tres capas como se puede apreciar en la figura.

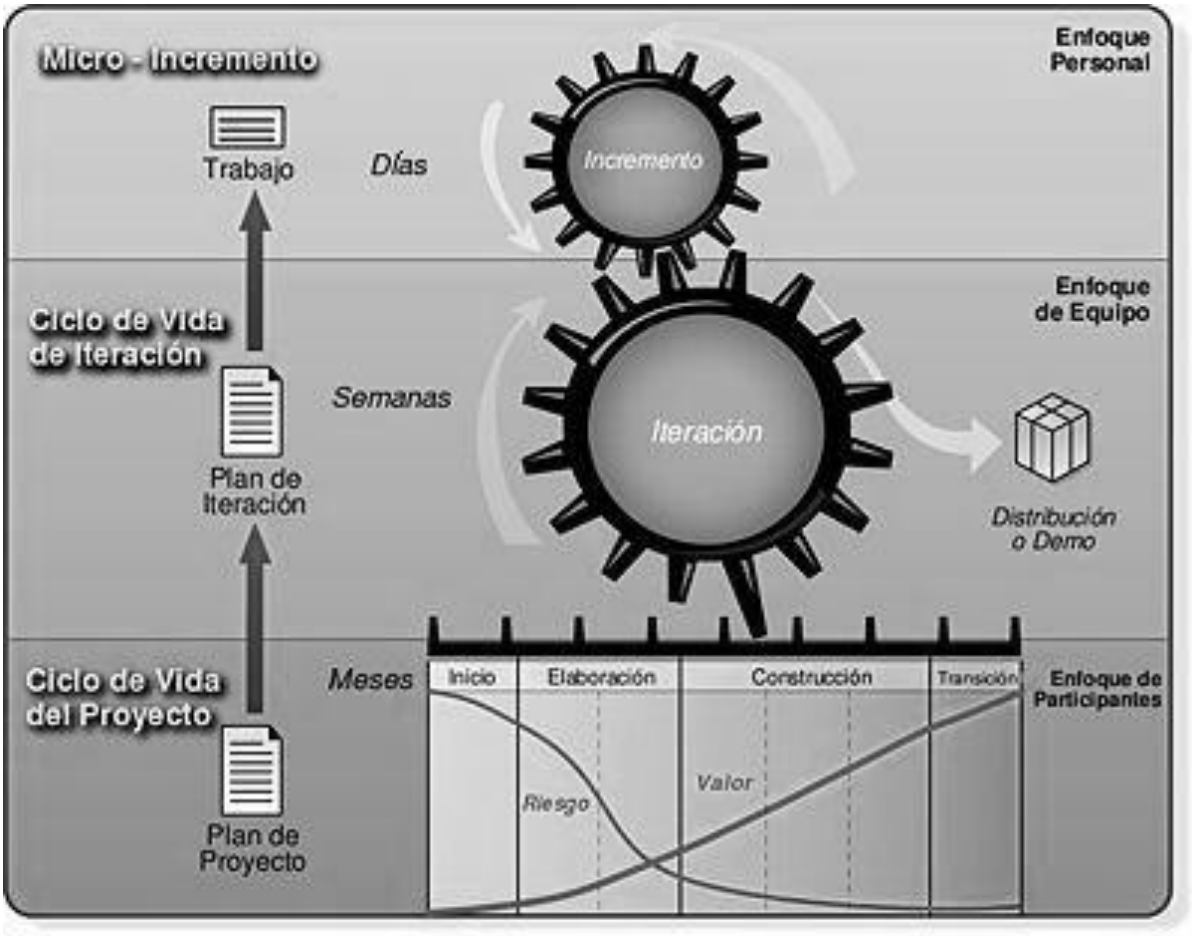

Figura 1. *OpenUP layers: micro-increments, iteration lifecycle and project lifecycle* (Yang, 2011)

Es recomendada para equipos pequeños que trabajan en un mismo local. Se enfoca en la reducción de riesgos a través de reuniones regulares y una implementación rigurosa de estrategias de mitigación.

Principios básicos de OpenUP (Balduino, 2007)

- Balancear las prioridades competitivas para maximizar el valor de los participantes del proyecto.
- Colaborar para alinear los intereses y compartir entendimiento.
- Evolucionar para obtener constantemente retroalimentación y mejorar.
- Enfocarse en la arquitectura tempranamente para minimizar riesgos y organizar el desarrollo.

Se selecciona dicha metodología porque está diseñada para equipos pequeños como es el caso, genera solo la documentación necesaria durante el proceso de desarrollo de software y permite la personalización de su proceso según la ocurrencia de nuevas necesidades y la generación de versiones. Además por la ventaja de que permite personalizar su proceso según las necesidades que aparecen a lo largo del ciclo de vida del software.

Lenguaje de modelado UML

Lenguaje unificado de modelado del inglés *Unified Modeling Language* (UML) es el más conocido y utilizado actualmente. Se utiliza para visualizar, especificar, construir y documentar los artefactos de un software. UML utiliza diagramas y una semántica bien definida para elaborar los artefactos del sistema en las distintas etapas de su ciclo de vida, fundamentalmente en el análisis y diseño (STEVENS, 2007). Brinda el lenguaje de modelado para:

- Modelo de proceso de negocios con casos de uso.
- Modelado de clases con objetos.
- Modelado de componentes.
- Modelado de distribución y despliegue.

El lenguaje ayuda al usuario a comprender bien el funcionamiento del software y reflexionar antes de invertir grandes cantidades de dinero en un proyecto que no le sea viable. El modelado ayuda además a mejorar la capacidad del equipo para gestionar la complejidad del software.

Herramienta CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora o *Computer Aided Software Engineering*) son aplicaciones informáticas destinadas al incremento de la productividad en el proceso de desarrollo de software. Reduciendo el costo de desarrollo de las aplicaciones en cuanto a términos de tiempo y dinero (SLIDESHARE, 2008).

Objetivos de las herramientas CASE:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.

- Mejorar los tiempos y costos de desarrollo y mantenimiento de sistemas informáticos.
- Mejorar la planificación.
- Automatizar el desarrollo del software, documentación, generación de código, pruebas de errores y gestión de proyectos.
- Reutilización de software, portabilidad y estandarización de la documentación.
- Gestión global de todas las fases de desarrollo de software.

Visual Paradigm 8.0

Herramienta que soporta el ciclo de vida completo de desarrollo de software. Soporta el lenguaje unificado de modelado (UML). Sencillo de utilizar, fácil de instalar y actualizar. Permite la generación de código a partir de diagramas para varios lenguajes como .Net, Java, PHP. Posibilita la representación gráfica de diagramas como: componentes, despliegue, secuencia, casos de uso; clase, actividad, estado, entre otros. Se integra con diversos sistemas gestores de bases de datos. Además permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando el lenguaje gráfico (LARMAN, 1999).

Websecurify Scanner 0.8

Es una solución avanzada para identificar con rapidez y precisión los problemas de seguridad de aplicaciones web. Websecurify ahorra tiempo al automatizar el tedioso proceso utilizado por los expertos para encontrar las vulnerabilidades. Funciona en sistemas operativos como Linux, Windows y Mac (WEBSEGURIFY, 2013).

1.6 Conclusiones parciales

Como resultado de la investigación realizada en este capítulo se puede concluir que:

- El estudio de los referentes teóricos permitió identificar que ninguna de las herramientas estudiadas brinda una solución al problema planteado.
- El estudio de las herramientas permitió detectar las funcionalidades básicas del módulo.
- Se identificaron las principales herramientas y tecnologías necesarias para el desarrollo del módulo.

Capítulo 2: Análisis y diseño del módulo para la detección de contenido duplicado

En este capítulo se presenta una propuesta general del módulo de detección de contenido web duplicado. Los requisitos funcionales (RF) y los requisitos no funcionales del módulo (RNF) identificados a partir del modelo conceptual construido, que generan los casos de uso del mismo (CU).

2.1 Descripción de la propuesta de solución

El módulo para la detección de contenido duplicado en portales web debe ser capaz de realizar la búsqueda de contenido duplicado interno en formato texto en los portales web desarrollados en Drupal 7, haciendo las comparaciones de similitud de los textos del portal a través de los algoritmos similitud coseno y similar text, permitiendo la personalización de su búsqueda a través de URL, título y cuerpo de página. Para brindar como resultado un reporte con los nodos que posean duplicidad parcial o total.

2.2 Modelo de dominio

El modelo de dominio es una representación de clases conceptuales significativas en un problema, no un diagrama que describe clases de software. Para consultar el modelo de dominio correspondiente acceder al anexo 1.

Tabla 1. Definición de conceptos del dominio

Concepto	Descripción
Acción	Acción que realiza un usuario sobre el sistema.
Sistema	Conjunto de partes que funcionan relacionándose entre sí con un objetivo preciso.
Grupo de usuarios	Grupo conformado por usuarios que poseen los mismos permisos.
Rol	Función que juega un usuario dentro del sistema.
Administrador	Permite realizar todas las acciones del sistema.
Editor	Rol con nivel de acceso limitado.
Usuario	Usuario que interactúa con el sistema.

El módulo presenta un sistema jerárquico que permite el acceso a sus funcionalidades solamente a los usuarios autorizados:

- **Editor**: puede consultar el reporte de contenido duplicado que devuelve el módulo en correspondencia por el metadato de búsqueda que fue seleccionado por el administrador.
- **Administrador**: puede acceder a todas las funcionalidades del módulo, incluidas la selección del metadato de búsqueda y la asignación de permisos.

2.3 Requisitos de software

Los requisitos de software son las cualidades que debe tener el producto de software. Ellos se dividen en dos grupos: los requisitos funcionales, que constituyen especificaciones detalladas de las funciones con que debe cumplir el producto y los requisitos no funcionales, son características del producto pero no forma parte de sus funcionalidades obligatorias.

2.3.1 Requisitos funcionales

- RF1. Configurar parámetros de búsqueda.
- RF2. Búsqueda de duplicado web por URL.
- RF3. Búsqueda de duplicado web por título.
- RF4. Búsqueda de duplicado web por cuerpo de página.
- RF5. Búsqueda de duplicado web por título y cuerpo de página.
- RF6. Asignar permisos.

2.3.2 Requisitos no funcionales

- **Soporte**
  - El módulo debe correr sobre los gestores de bases de datos MySQL y PostgreSQL.
  - Fácil instalación.
  - El módulo debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.
- **Usabilidad**
  - Debe poseer una interfaz intuitiva y agradable al usuario.
  - El módulo podrá ser utilizado por personas con pocos conocimientos de informática.
  - Internacionalización.
- **Seguridad**
  - Se define el acceso al módulo y sus funcionalidades mediante la asignación de permisos por roles de usuarios.
  - La carpeta donde se encuentre el portal solo tendrá permiso de lectura.
  - Los errores deben mostrar la menor cantidad de detalles posible, para evitar brindar información que comprometa la seguridad e integridad del sistema.

- **Legales:**
  - El módulo está basado en la licencia GNU/GPL versión 2.

- **Fiabilidad**
  - Los datos obtenidos de las búsquedas deben estar en correspondencia con los algoritmos empleados para calcular la similitud.
- **Apariencia o interfaz externa**
  - El portal debe ser compatible con los navegadores Chrome, Firefox, Safari, Opera e Internet Explorer a partir de su versión 9.
  - El sistema debe poseer un diseño web adaptable.

2.4 Definición de casos de uso

A continuación se definen los casos de uso que se corresponden con los requisitos identificados, los cuales serán una de las bases durante las fases de análisis, diseño e implementación del módulo. Para consultar la descripción de los casos de uso consultar el anexo 2.

Tabla 2. Casos de uso del sistema

CU	Referencia
CU1: Asignar permisos	RF6
CU2: Configurar parámetros de búsqueda	RF1
CU3: Realizar búsqueda de duplicado web por metadato específico	RF2, RF3, RF4, RF5

2.4.1 Diagrama de casos de uso

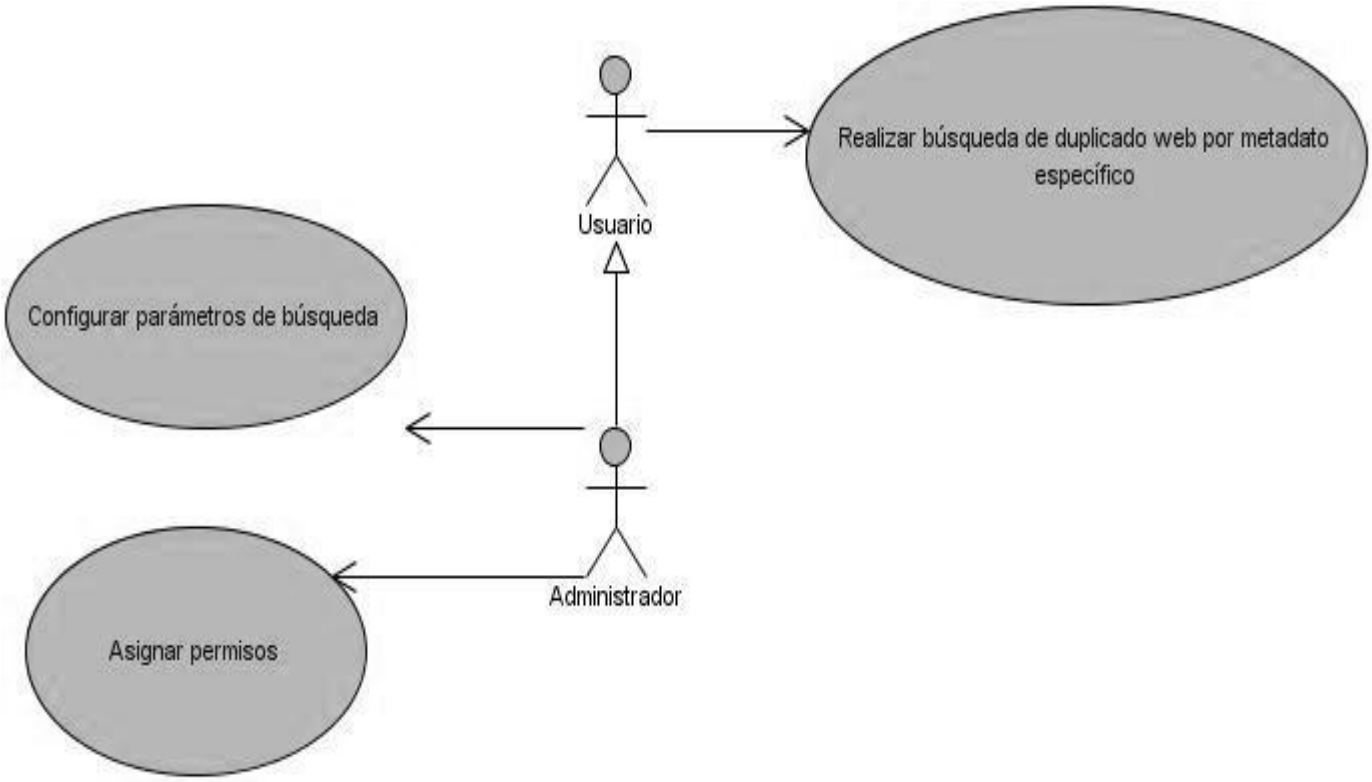

Figura 2. Diagrama de casos de uso

2.5 Arquitectura de Drupal

Drupal posee una arquitectura estructurada en diferentes capas, por lo que para el desarrollo del módulo se propone el trabajo con la arquitectura n-capas, específicamente 5 capas que son descritas a continuación (Vandyk, 2011):

Plantillas (*templates*): Esta capa establece la apariencia gráfica que se le muestra al usuario. Esta separación entre la información y los estilos permite cambiar la apariencia del portal web sin necesidad de modificar los contenidos.

Vistas (*views*): Es la capa encargada de mostrar en los temas los cambios realizados a través de los módulos.

Entidades (*entities*): Representa las entidades, que engloban los nodos, los usuarios, las taxonomías y los comentarios. Esta nueva estructura permite que sea posible añadirle campos a todo aquello que sea una entidad.

Módulos (*modules*): Engloba los elementos que operan sobre los nodos otorgando funcionalidades a Drupal. Permiten incrementar sus capacidades o adaptarlas a las necesidades de cada portal web.

Base de Datos (*database*): Esta capa es la encargada de gestionar el acceso a la información almacenada referente al funcionamiento del sistema y a los contenidos que serán mostrados a través del tema activo.

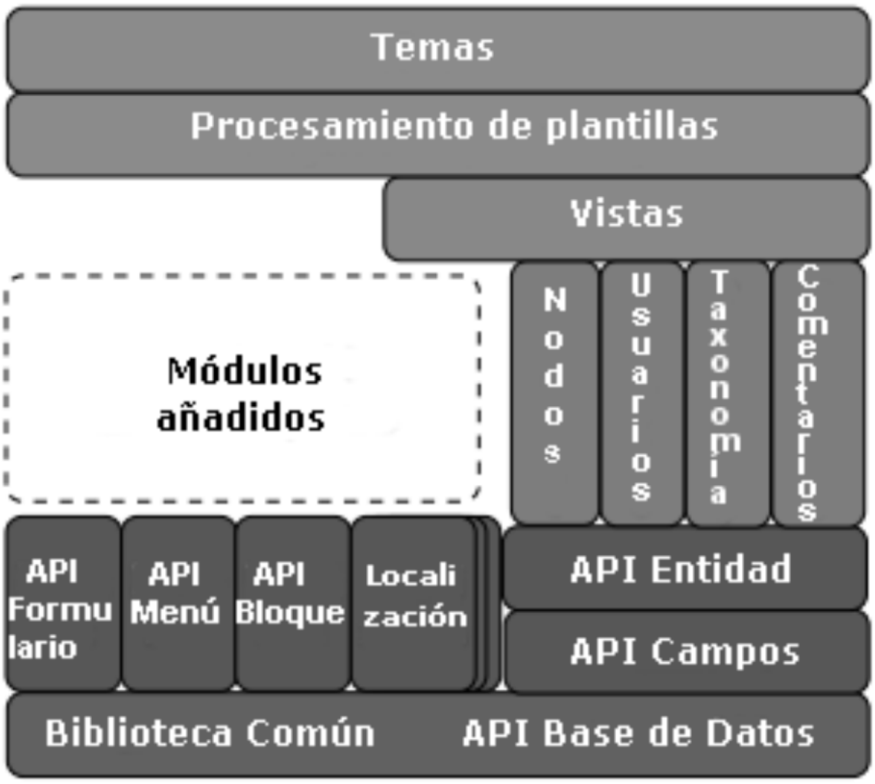

Figura 3. Arquitectura de Drupal 7. Estructura en 5 capas (Vandyk, 2011)

El módulo para la detección de contenido duplicado se encuentra ubicado dentro de la capa Módulos de la arquitectura de Drupal 7, pero se interrelaciona con las restantes capas. El módulo hace uso de la capa

Base de Datos (BD), específicamente de su API para realizar consultas que sean independientes al gestor utilizado. Además de su interacción con la capa Entidades cuando crea sus permisos de acceso.

2.6 Patrones de diseño en Drupal

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

Observer **(Observador)**

Los módulos que implementan un *hook* de inserción o actualización de una entidad, son declarados como observadores de las entidades con las que interactúan. En el caso del módulo desarrollado, el patrón se evidencia en la inserción de los permisos *Administrator Access to duplicate detector* y *Common user access to duplicate detector*.

Bridge **(Puente)**

La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón puente. El módulo implementado es programado de manera que sea independiente del motor de base de datos que utiliza el sistema. Esto se logra por la capa de abstracción de base de datos, sobre la que se pueden desarrollar consultas siguiendo la API definida.

Este patrón se evidencia en el módulo detector de contenido duplicado que fue programado logrando independencia del motor de BD que se utilice. Esto se evidencia en el *hooks* detector_settings () que permite conectarse a la BD y realizar consultas abstrayéndose del SGBD que se emplee para manejar los datos almacenados.

Chain of responsability **(Cadena de responsabilidades)**

El sistema de menú de Drupal es la evidencia del patrón cadena de responsabilidades. En cada petición de una página, el sistema de menú de Drupal determina si hay algún módulo para responder la petición, si el usuario tiene acceso al recurso solicitado y que función se debe llamar para procesar la petición. En este proceso se trasmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos. En el módulo para la detección de contenido duplicado mediante el hook detector_menu() en dependencia de la petición recibida por el módulo, se delega la responsabilidad a la función encargada de atenderla.

2.7 Modelos de diseño

2.7.1 Diagramas de clases del diseño

Un diagrama de clases de diseño (DCD) representa las especificaciones de las clases e interfaces software en una aplicación. Las clases de diseño de los DCD muestran las definiciones de las clases software en lugar de los conceptos del mundo real (Larman, 2003).

A continuación se observa el diagrama de clases del diseño del CU # 3: Realizar búsqueda de duplicado web por metadato específico. Los restantes se pueden consultar en el anexo 3.

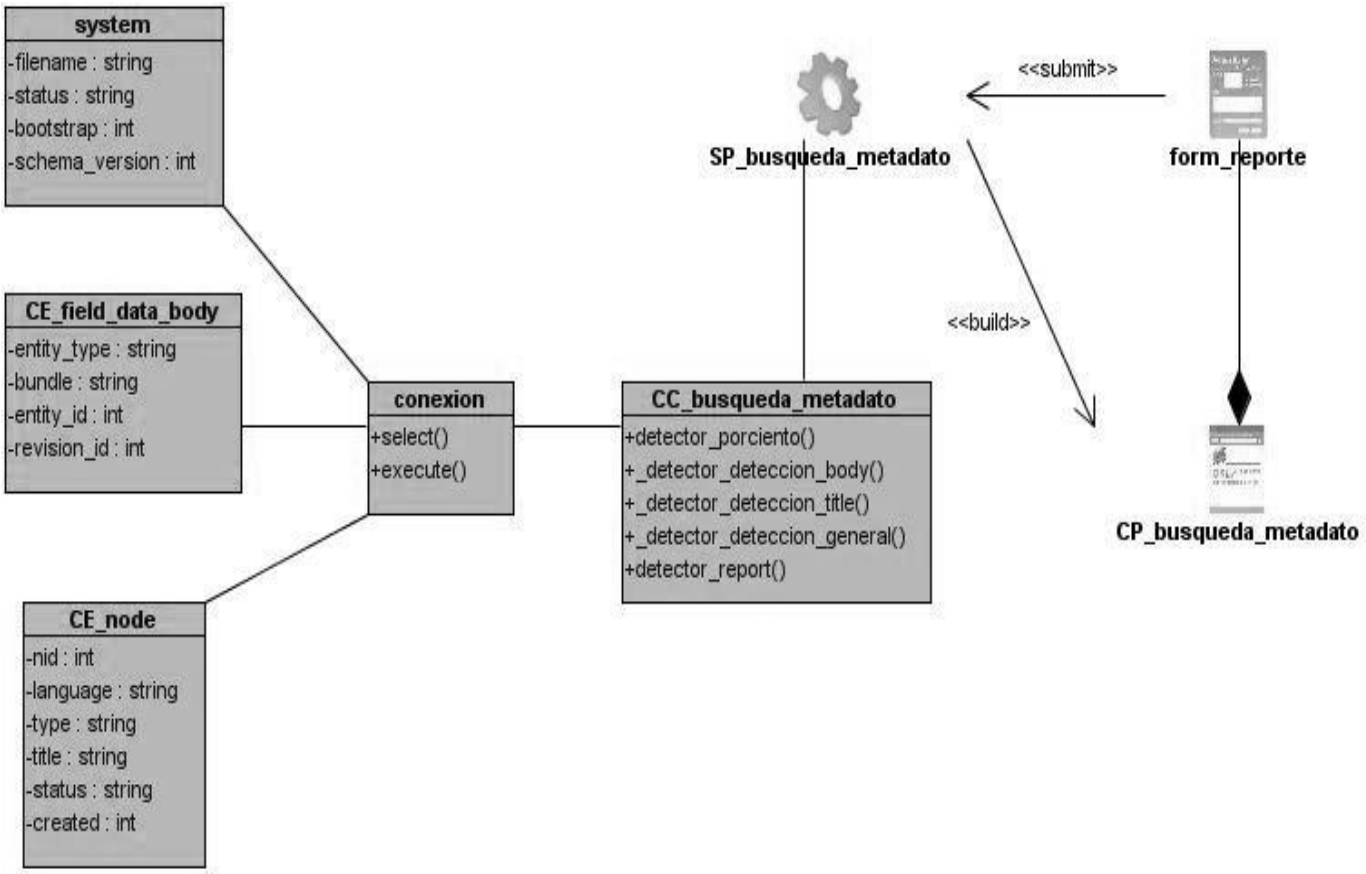

Figura 4. Diagrama de clases del diseño: Realizar búsqueda de duplicado web por metadato específico

2.7.2 Diagramas de secuencia del diseño (DS)

Ilustran las interacciones en un tipo de formato con aspecto de una valla, en el que cada objeto nuevo se añade a la derecha, mostrando claramente la secuencia y ordenación en el tiempo de los mensajes con una notación simple (Larman, 2003).

A continuación se observa el diagrama de secuencia: Realizar búsqueda de duplicado web por metadato específico para el escenario título. Los restantes se pueden consultar en el anexo 4.

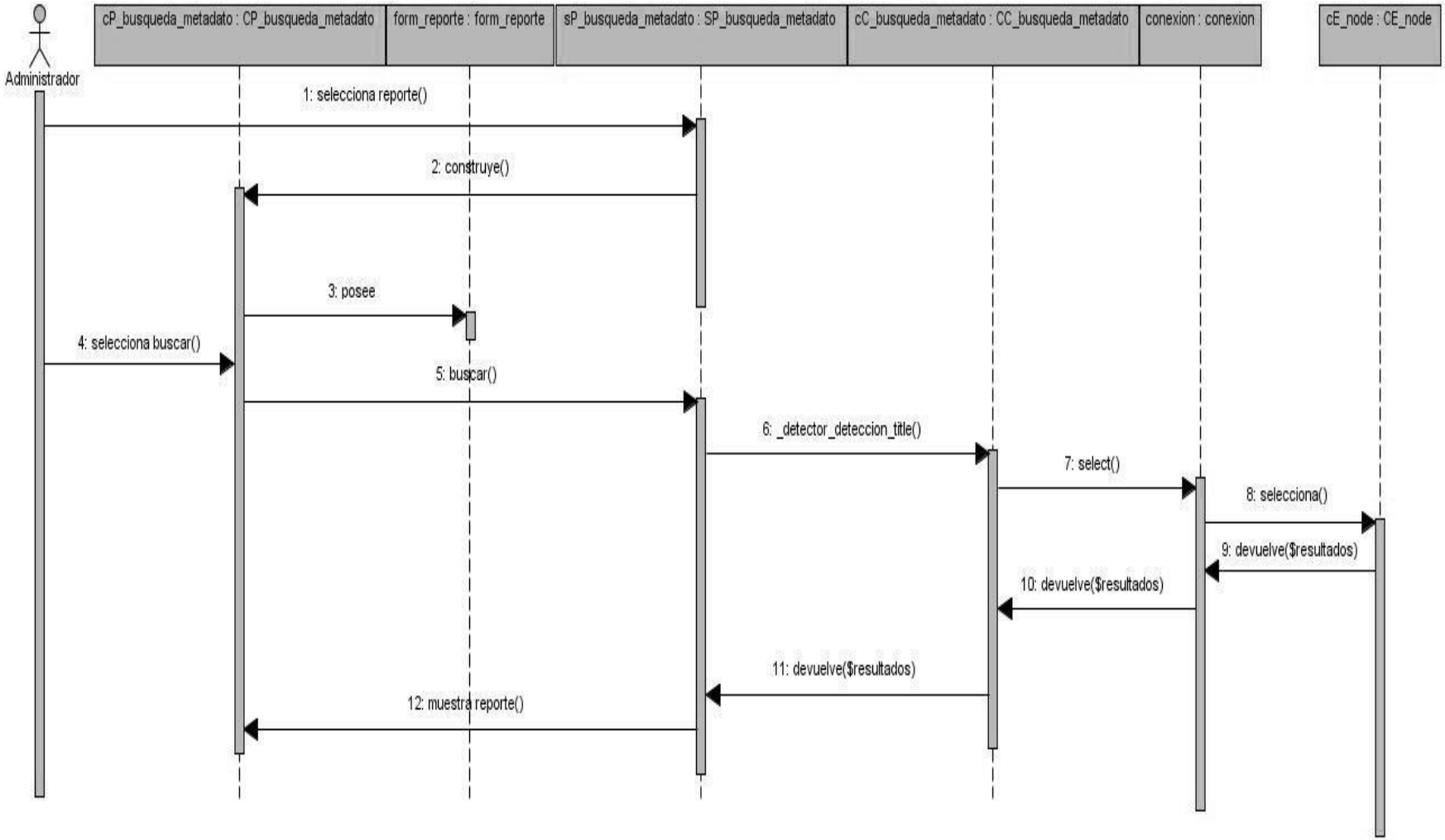

Figura 5. Diagrama de secuencia: Realizar búsqueda de duplicado web por metadato específico (escenario título)

2.8 Conclusiones parciales

El desarrollo de este capítulo permitió:

- Los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, sirvieron de guía para desarrollar las distintas funcionalidades y de este modo satisfacer las necesidades detectadas.
- Los artefactos generados según la metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.

Capítulo 3: Implementación y prueba del módulo para la detección de contenido duplicado

En el presente capítulo se presentan los componentes y los estándares de codificación que sustentan la implementación del módulo para la detección de contenido duplicado en portales web. Se describe y fundamenta el proceso de validación de la solución implementada, mediante la utilización de los casos de pruebas y la herramienta de seguridad.

3.1 Diagrama de componentes

El diagrama de componentes muestra los componentes de un sistema de software conectados por las relaciones de dependencias lógicas entre cada uno de ellos. Provee una vista arquitectónica de alto nivel del sistema, ayudando a los desarrolladores a visualizar el camino de la implementación. Cada componente representa una unidad del código (fuente, binario o ejecutable), que permite mostrar las dependencias en tiempo de compilación y ejecución. La realización del diagrama posibilita tomar decisiones respecto a las tareas de implementación y los requisitos (Rivera, 2008).

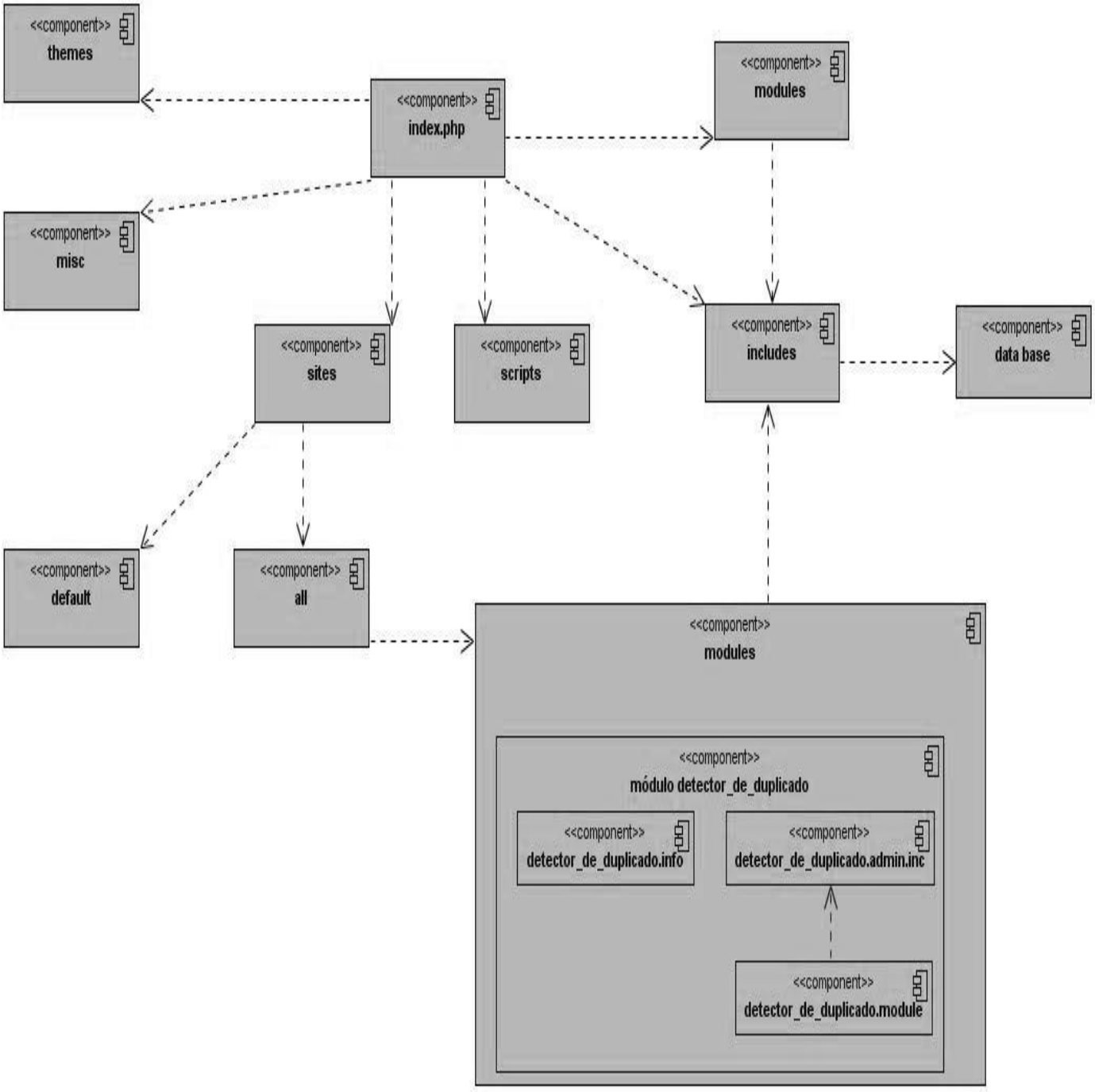

Figura 6. Diagrama de componentes del módulo para la detección de contenido duplicado en portales web

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

Tabla 3. Descripción de componentes

COMPONENTES	DESCRIPCIÓN
index.php	Este componente es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS Drupal.
themes	En este componente se incluyen los temas que vienen con la distribución de Drupal.
misc	Este componente incluye archivos javascript e imágenes requeridas por el sistema.
includes	Este componente tiene un conjunto de ficheros indispensables para el funcionamiento del CMS Drupal.
scripts	Contiene utilidades adicionales que no utiliza Drupal directamente, pero que se pueden utilizar desde la línea de comandos de *shell*. Por ejemplo, el script password-hash.sh permite obtener una contraseña codificada a partir de la contraseña original.
modules	En este componente se encuentran los módulos básicos del núcleo de Drupal.
sites	La carpeta sites contiene los extras y modificaciones que se añaden a la distribución original. En esta se encuentran los módulos adicionales creados, añadidos o descargados del repositorio de módulos de Drupal, colocados en sites/all/modules. La carpeta incluye tras la instalación el archivo de configuración del portal (settings.php).

.module	Archivo que contiene las llamadas a las funciones ganchos de Drupal.
.info	Contiene información básica sobre el módulo (nombre, descripción, versión de Drupal, requisitos mínimos, posibles dependencias, paquete al que pertenece el módulo y ficheros que incluye).
.admin.inc	Archivo donde se implementan las funciones que posee el módulo.

3.1.1 Modelo de despliegue

Representa de forma visual las relaciones físicas que existen entre los componentes de software y hardware en el sistema. Los nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (Jacobson, 2000). En la siguiente figura se muestra el diagrama de despliegue correspondiente a la solución propuesta.

Figura 7. Diagrama de despliegue

PC: Estación de trabajo que presenta un navegador web para conectarse a los portales web hospedados en el servidor web utilizando el protocolo de comunicación HTTPS, que además tienen instalado el módulo para la detección de contenido web duplicado.

Servidor web: Estación de trabajo que hospeda el código fuente del módulo y que le brinda a los usuarios las interfaces del mismo para realizar los procesos definidos para cada uno de los roles. Esta estación se comunica con el servidor de BD donde se almacenan los datos de los portales realizando la comunicación mediante el protocolo TCP/IP.

Servidor de BD: Servidor encargado del almacenamiento de los datos de los portales. Se comunica con el servidor web.

3.2 Estándares de codificación

Los estándares de codificación permiten que el código generado durante la implementación de un proyecto se pueda leer con una mayor facilidad. Además permiten la comprensión y modificación del código independientemente de su autor permitiendo además que el mismo presente una alta calidad y no presente errores. Drupal brinda a sus desarrolladores las siguientes normas de codificación (Gil, 2012).

Indentación

La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos.

En Drupal se debe indentar con 2 espacios, nunca con tabuladores. Además, no se debe dejar espacios en blanco al final de cada línea. En el siguiente ejemplo se muestra un fragmento de código con las indentaciones realizadas, de 2 espacios cada una, y los saltos de línea o *enter* al final de cada línea (sin dejar espacios).

```
function detector_menu() {
  $items['admin/reports/reporte_deduplicado'] = array(
    'title callback' => '_detector_simple_title_callback',
    'title arguments' => array(t('Duplicate Content Detector: ')),
    'description' => t('Duplicates contents'),
    'page callback' => 'drupal_get_form',
    'page arguments' => array('detector_settings'),
    'access callback' => 'user_access',
    'access arguments' => array('access user_detector'),
    'file' => 'detector.admin.inc',
  );
  return $items;
}
```

Etiquetas de apertura y cierre de PHP

Cuando se escribe en PHP, siempre se deben utilizar las etiquetas <?php y ?>, y en ningún caso la versión corta <? y ?>. En general se omite la etiqueta de cierre de PHP (?>) al final de los archivos .module y .inc. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre ?>), que serían identificados como salida HTML y podrían provocar un error muy típico, "*Cannot modify header information - headers already sent by...*".

Por tanto, la etiqueta de cierre final del archivo (?>) es opcional en Drupal.

No hay que confundir esto con el uso normal del lenguaje PHP en archivos que también contienen HTML (como por ejemplo los archivos de plantilla .tpl.php), donde cada fragmento de PHP debe llevar sus correspondientes etiquetas de apertura y cierre, para diferenciarlo del código HTML.

Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, $numero = 3, en lugar de $numero=3. Esto se aplica a operadores como +, -, *, /, =, ==, !=, >, <, . (Concatenación de cadenas), .=, +=, -=, etc.

Los operadores unarios como ++, -- no deben tener separación. Por ejemplo, $numero++.

Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres.

Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<h1>$title</h1>". También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.

Uso de punto y coma (;) en código PHP

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo <?php print $title ?>. En Drupal es siempre obligatorio: <?php print $title; ?>.

- Correcto: <?php print $title; ?>

- Incorrecto: <?php print $title ?>

Estructuras de control

Con respecto a las estructuras de control, hay que tener en cuenta las siguientes normas:

- Debe haber un espacio entre el comando que define la estructura (*if, while, for*, etc.) y el paréntesis de apertura. Esto es así para no confundir las estructuras de control con la nomenclatura de las funciones.
- La llave de apertura { se situará en la misma línea que la definición de la estructura, separada por un espacio.
- Se recomienda usar siempre las llaves {} aún en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control).
- Las estructuras *else* y *elseif* se escribirán en la línea siguiente al cierre de la sentencia anterior.

Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo o tema, para evitar así duplicidad de funciones.

En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior. Ejemplo:

function forum_help($path, $arg) {

En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros, como se muestra en el siguiente ejemplo:

$field = field_info_instance('node', 'taxonomy_forums', $node->type);

Como excepción, es posible usar más de un espacio antes de una asignación (=) para mejorar la presentación, cuando se estén realizando varias asignaciones en bloque:

$numero1 = foo($a, $type);

$primer_valor = foo2($b);

$i = foo3();

Arrays

Los valores dentro de un array (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados.

Cuando la línea de declaración del array supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

$vector1 = array(1, 2, 'clave' => 'valor');

$vector2 = array(

'forum' => 'foro1',

'template' => 'forums',

'arguments' => array('tid' => NULL, 'topics' => NULL),

'size' => 128,

);

Nombres de módulos

El nombre de un módulo nunca debería incluir guiones bajos, aunque se componga de varias palabras. De esta forma es más fácil identificar el módulo al que pertenece una función, ya que el prefijo o nombre del módulo es todo aquello que esté antes del primer guión bajo. Por ejemplo, es aconsejable utilizar mimodulo en lugar de mi_modulo. Esta regla no es obligatoria.

Nombres de archivos

Los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación, que tendrán extensión .txt y el nombre en mayúsculas. Por ejemplo, README.txt, INSTALL.txt, etc.

Comentar el código

En este apartado se debe diferenciar entre los comentarios para aclarar determinados fragmentos de código, que se insertan en cualquier punto del mismo, y los comentarios de documentación.

Los comentarios de documentación suelen escribirse al principio de un archivo o de cada función y se utilizan para generar documentación de ayuda a través de aplicaciones que extraen la información a partir de las etiquetas empleadas.

En el primero de los casos se pueden utilizar las etiquetas /* */ para comentarios en varias líneas y // para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, ésta deberá escribirse en mayúsculas (por ejemplo TRUE o FALSE).

3.3 Validación de la propuesta de solución

Las pruebas de software son procedimientos realizados para verificar la calidad de un software y pueden ser aplicadas periódicamente. Estas tienen como objetivo identificar posibles errores en la aplicación. Existen varias estrategias de pruebas que suelen ser utilizadas, dentro de las que se pueden mencionar:

Pruebas de Caja Blanca

Dirigidas a las funciones internas del sistema. Consisten en una verificación técnica del software que los desarrolladores pueden usar para examinar si su código trabaja como es esperado. Se realizan probando la lógica de la aplicación y comprobando el estado del software en varios puntos, verificando que el resultado de dicho estado coincida con lo esperado.

Pruebas de Caja Negra

Se desarrollan sobre la interfaz visual del software y se centran en los requisitos funcionales de la aplicación, sin adentrarse en el funcionamiento interno de la aplicación. A través de su realización se pueden encontrar errores de interfaz, funciones incorrectas, errores de salida y problemas con el acceso a datos.

Pruebas de Sistema

Las pruebas de sistema tienen como objetivo verificar el sistema de software para comprobar si el mismo cumple con sus requisitos. Cuenta con distintos tipos de pruebas, las cuales algunas son funcionales, de usabilidad, de rendimiento, de seguridad, entre otras.

A continuación se presentan los resultados de las pruebas de caja negra obtenidos luego de aplicarlas al módulo.

3.3.1 Funcionales

Las pruebas de funcionalidad se realizan con el objetivo de garantizar el funcionamiento adecuado de los requisitos funcionales. Para su realización es necesario diseñar un conjunto de casos de prueba que son utilizados para someter a las diferentes funcionalidades del módulo y de este modo verificar su correcta ejecución.

Tabla 4. Muestra para las pruebas

Nodos	Cantidad
Duplicación exacta	20
Duplicación parcial	30
Únicos	15

Para la realización de las pruebas funcionales se tomó como muestra a tener en cuenta la presentada en la tabla 4. Obteniendo como resultado un correcto funcionamiento de las funcionalidades del módulo. Para visualizar los casos de prueba se puede consultar el anexo 5.

Resultados de las pruebas funcionales

Se probaron todos los casos de prueba que responden a las funcionalidades del módulo para la detección de contenido duplicado en portales web. En total fueron detectadas 25 no conformidades de las cuales todas fueron resueltas, los principales errores detectados se correspondían a errores de redacción, ortográficos y de idioma.

A continuación se muestra una gráfica donde se desglosan las no conformidades detectadas en las tres iteraciones realizadas. En la primera iteración se recoge un total de 15 no conformidades de las cuales todas fueron resueltas, la segunda iteración arrojó un total de 10 no conformidades siendo completamente solucionadas y para una tercera iteración y final no se detectaron no conformidades.

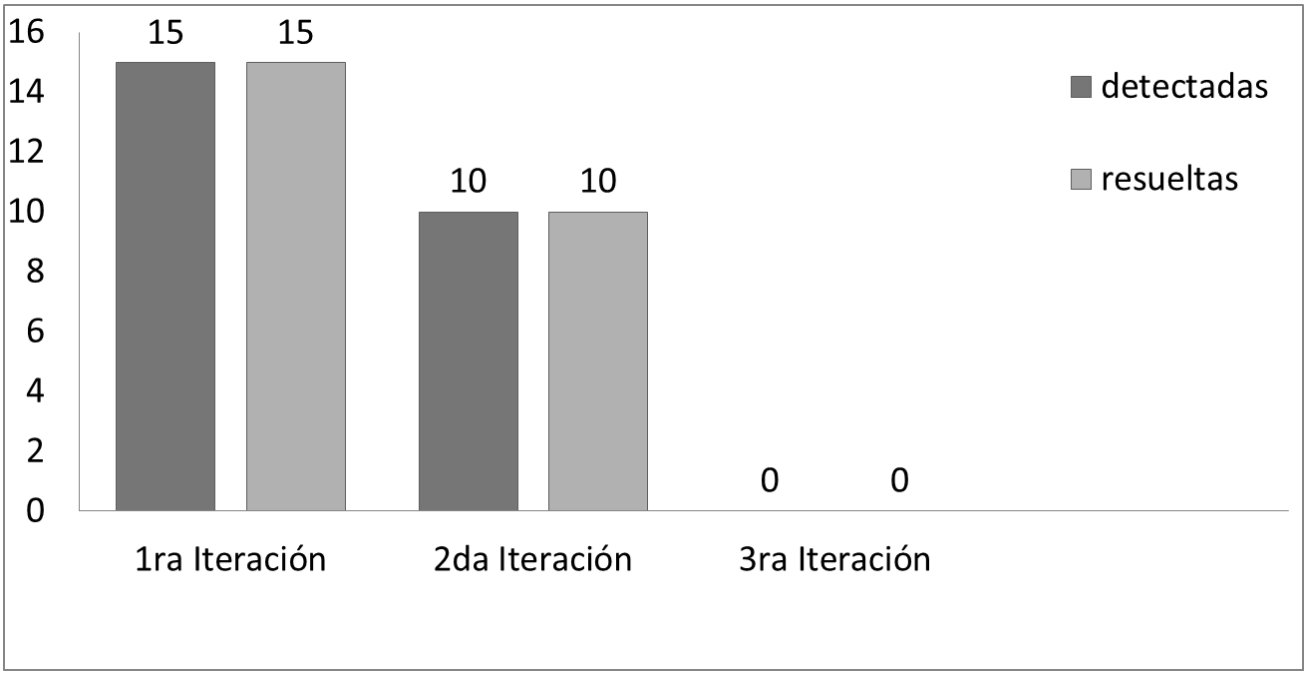

Figura 8. Resultados de las pruebas funcionales

3.3.2 Integración

Las pruebas de integración se realizan para confirmar que el módulo se integra de forma correcta con el sistema. Para la realización de ellas se probó el módulo en portales desarrollados con distintos gestores de bases de datos como el portal web de Consultoría Jurídica Internacional desarrollado utilizando como gestor de bases de datos MySQL y el portal web de la revista Alma Mater que se encuentra desarrollado con el gestor PostgreSQL. El desarrollo de las pruebas garantizó que el módulo presentaba un correcto funcionamiento en ambos portales y que las funcionalidades de los portales no eran afectadas.

3.3.3 Seguridad

Las pruebas de seguridad se aplican para garantizar que los usuarios solo puedan acceder a las funcionalidades a las que poseen permiso de acceso. Constituye una de las principales pruebas pues permiten evitar ataques de seguridad a través de vulnerabilidades existentes en el módulo.

Se realizaron pruebas de seguridad haciendo uso de las aplicaciones Websecurify Scanner 0.8 que es una solución avanzada para identificar con rapidez y precisión los problemas de seguridad de aplicaciones web. Ahorra tiempo al automatizar el tedioso proceso utilizado por los expertos para encontrar las vulnerabilidades. Funciona en sistemas operativos como Linux, Windows y Mac (WEBSEGURIFY, 2013).

A continuación se muestran las vulnerabilidades identificadas con el uso de la herramienta seleccionada:

Tabla 5. Pruebas de seguridad (Websecurify Scanner)

Vulnerabilidad	Descripción	Número de ocurrencias
Inyección SQL	Técnica de inyección de código que explota una vulnerabilidad de seguridad que ocurre en la capa de base de datos de una aplicación web. La vulnerabilidad estaba presente cuando la entrada del usuario estaba mal filtrada por caracteres de escape de cadenas literales incorporados en SQL.	1
Petición falsificada	Ataque mediante el cual se transmiten comandos no autorizados de un usuario en el que confíe la aplicación. Explota la confianza que tiene un portal en el navegador del usuario.	1

Divulgación de ruta	El servidor o aplicación da a conocer rutas del sistema. Esta información podría ser utilizada por atacantes para conocer la estructura de datos de la aplicación y acceder a información privada.	1

Los errores detectados se analizan a través de enlaces que proporciona la aplicación para dar a conocer la ubicación de los mismos y de este modo facilitar su solución. Las vulnerabilidades detectadas por la herramienta correspondían a falsos positivos.

3.4 Interfaces de la solución

Figura 9. Búsqueda por título

Figura 10. Búsqueda por URL

3.5 Conclusiones parciales

En el capítulo se realizó el análisis de los tipos de pruebas realizados al módulo. Se confeccionaron casos de prueba con el fin de comprobar el funcionamiento del mismo. Obteniendo como resultado las siguientes conclusiones:

- La elaboración de los diagramas y la descripción de los componentes que lo conforman, permitió una mejor comprensión de la estructura del módulo implementado.
- Aplicar los estándares de codificación permitió obtener en el módulo para la detección de contenido duplicado en portales web un código legible, estándar y fácil de comprender.

- El proceso de validación de la solución propuesta permitió demostrar la calidad del módulo desarrollado.

Conclusiones

Con la investigación realizada se logró el desarrollo del módulo de detección de contenido duplicado en portales web, capaz de realizar la detección de contenido duplicado internos existentes en un portal web desarrollado en Drupal 7. El mismo perseguía resolver la detección del duplicado web existente en los portales desarrollados en dicho sistema de administración de contenido. Los objetivos propuestos fueron cumplidos con satisfacción, generando cada uno de ellos los siguientes resultados:

- La elaboración del marco teórico permitió identificar la necesidad de crear el módulo para la detección de contenido duplicado en portales web, determinando la selección de las herramientas, metodología y tecnologías factibles a utilizar en el desarrollo del módulo.
- Los artefactos generados durante el flujo de trabajo de análisis y diseño sirvieron para conformar la primera visión de la implementación del sistema, defendiéndose durante la misma una estrategia para lograr una construcción flexible y robusta a través de la utilización de una arquitectura afín, patrones de diseño bien establecidos y estándares de codificación.
- Mediante la realización de pruebas de software para evaluar las funcionalidades se probó la conformidad de los requisitos especificados y se comprobó el cumplimiento del objetivo general trazado.

Recomendaciones

A modo de recomendación se propone:

- Aplicar algoritmos mejorados para calcular la similitud entre los contenidos del portal web.
- Realizar el cálculo de la similitud en un servidor externo al servidor donde está publicado el portal web.
- Integrar la recopilación de información al buscador Orión para realizar detección de contenido duplicado externo.

Bibliografía

ADAPTPARTNERS.COM. SEO Risk Assessment. 2015. Available from Internet:<http://adaptpartners.com/risk-audit-and-prevention-from-adapt-partners/>.

ADAPTPARTNERS.COM. Technical SEO. 2015. Available from Internet:<http://adaptpartners.com/technical-seo/>.

AMÓN, I., F. MORENO AND J. ECHEVERRI. ALGORITMO FONÉTICO PARA DETECCIÓN DE CADENAS DE TEXTO DUPLICADAS EN EL IDIOMA ESPAÑOL. In Revista Ingenierías Universidad de Medellín. 2012, vol. 11, p. 127-138.

ANONYMOUS. Casi todas las herramientas SEO que podrías desear para tu web. 2014. Available from Internet:<https://aqua2webs.wordpress.com/2014/07/21/casi-todas-las-herramientas-seo-que-podrias-desear-para-tu-web/>.

ANONYMOUS. 7 herramientas gratuitas para detectar contenido duplicado. 2014. Available from Internet:<http://bloggertrucos.com/7-herramientas-gratuitas-para-detectar-contenido-duplicado/>.

AROCHE, J. Google nos habla de las mejores prácticas SEO. 2008. Available from Internet:<http://www.maestrosdelweb.com/google-nos-habla-de-las-mejores-practicas-seo/>.

ALVAREZ, S. Sistemas gestores de bases de datos. 2014. Available from http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html.

BALDUINO, R. Introduction to OpenUP (Open Unified Process). In., 2007.

BENAVIDES, M. S. K. R. Algoritmos de Similaridad y Distancia. In.

BAST, H. AND M. CELIKIK. Efficient Fuzzy Search in Large Text Collections. In ACM Transactions on Information Systems. 2010, vol. 9.

BUSINESSONLINE.COM. Enterprise SEO Solutions – Our Perspective. 2015. Available from Internet:<http://www.businessol.com/solutions/organic-search/>.

CAMBIASO, D. 9 Servicios web para detectar contenido plagiado. 2010. Available from Internet:<http://pixelcoblog.com/9-servicios-web-para-detectar-contenido-plagiado/>.

CAMPOS, O. Breve introducción a CSS3. 2011. Available from Internet:<http://www.genbetadev.com/desarrollo-web/breve-introduccion-a-css3>.

CARABAÑO, E. Cómo solucionar los problemas SEO de contenido duplicado. 2014. Available from Internet:<http://www.bluecaribu.com/como-solucionar-los-problemas-de-contenido-duplicado/>.

CHRISTIANSSON, B., M. FORSS AND I. HAGEN. GoF Design Patterns - with examples using Java and UML2. 2008. Available from Internet:<http://www.usp.br/thienne/coo/material/GoFDesignPatterns.pdf>.

CLEMENTE, E. Como Detectar el Contenido Duplicado Interno. 2014. Available from Internet:<http://okhosting.com/blog/como-detectar-el-contenido-duplicado-interno/>.

COHEN, W.W. Integration of Heterogeneous Databases without Common Domains Using Queries Based on Textual Similarity. *Proceedings of the SIGMOD International Conference Management of Data SIGMOD'98.* 1998, pp. 201-212.

COHEN, W.W., RAVIKUMARAND, P. FIENBERG, S.E. A Comparison of String Distance Metrics for Name-Matching Tasks. *International Joint Conference on Artificial Intelligence.* 2003, pp. 73-78.

COHEN, W. W., P. RAVIKUMAR AND S. E. FIENBERG A Comparison of String Metrics for Matching Names and Records. American Association for Artificial Intelligence, 2003.

DRUPAL.ORG. Drupal CMS Benefits. Available from Internet:<https://www.drupal.org/features>.

DRUPAL.ORG. Coding standards. 2015. Available from Internet:<https://www.drupal.org/coding-standards>.

ENACHE, M. C. OPTIMIZATION METHODS AND SEO TOOLS. In International Conference "Risk in Contemporary Economy". Galati, Romania, 2014.

FLORIDO, M. Contenido duplicado – 10 Herramientas para detectarlo. 2014, no. 4. Available from Internet:<http://www.marketingandweb.es/marketing/contenido-duplicado/>.

GALÁN, S. M. Filtrado Colaborativo y Sistemas de Recomendación. In IRC'07. Leganés, Madrid, Spain, 2007.

GARCÍA, J. F. Métricas de Similitud para Búsqueda Aproximada. In Revista de Tecnología. 2007, vol. 6.

GIL, F. Experto en Drupal 7 [online]. 2012. Available from World Wide Web:<https://www.forcontu.com>.

GOOGLE. Contenido duplicado. 2014. Available from Internet:<https://support.google.com/webmasters/answer/66359?hl=es>.

GOOGLE. Contenido copiado. 2015. Available from Internet:<https://support.google.com/webmasters/answer/2721312?hl=es>.

GOTOH, O. *An Improved Algorithm for Matching Biological Sequences*. *Journal of Molecular Biology*. 162 (3). 1982, pp. 705-708.

GUPTA, T. AND L. BANDA A HYBRID MODEL FOR DETECTION AND ELIMINATION OF NEAR-DUPLICATES BASED ON WEB PROVENANCE FOR EFFECTIVE WEB SEARCH. International Journal of Advances in Engineering & Technology, 2012.

HASSANZADEH, O., F. CHIANG AND H. C. LEE. Framework for Evaluating Clustering Algorithms in Duplicate Detection. In VLDB. 2009.

HERNÁNDEZ, J. D. P. AND R. C. V. HERNÁNDEZ. Sistema para la detección de plagio y validación de estructura en los documentos científicos emitidos en el Centro de Identificación y Seguridad Digital (CISED). 2012. Available from Internet:<http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_05267_12/1/TD_05267_12.pdf>.

HEUSER, C. A., F. N. A. KRIESER AND V. M. ORENGO. SimEval - A Tool for Evaluating the Quality of Similarity Functions. In Twenty-Sixth International Conference on Conceptual Modeling. Auckland, New Zealand, 2007, vol. 83.

HUNT, F. Total number of Websites & Size of the Internet as of 2013 2013. Available from Internet:<http://www.factshunt.com/2014/01/total-number-of-websites-size-of.html>.

INTERNETWORLDSTATS. Internet Users in the World. 2014. Available from Internet:<http://www.internetworldstats.com/stats.htm>.

JACOBSON, J. 2000. El proceso unificado de desarrollo de Software. Madrid: PEARSON EDUCACIÓN, S.A., 2000.

JARO, M.A. *Unimatch: A Record Linkage System User's Manual, technical report.* Washington. D.C.: US Bureau of the Census. 1976.

KIRUTHIKA, M., S. DANGE AND P. SANDHYA Title Based Duplicate Detection of Web Documents. International Journal of Electronics and Computer Science Engineering, 2014, 1.

LARMAN, C. UML y Patrones. 2003, vol. 2. Available from Internet:<http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDiseno.pdf>.

LARMAN, C. UML y patrones: introducción al análisis y diseño orientado a objetos. ln., 1999.

LUCIA. Las 67 mejores herramientas gratuitas de SEO. 2014. Available from Internet:<http://blog.bitmarketing.es/las-67-mejores-herramientas-gratuitas-de-seo/>.

LUZ, S. D. Criptografía : Algoritmos de autenticación (hash). 2010. Available from Internet:<http://www.redeszone.net/2010/11/09/criptografia-algoritmos-de-autenticacion-hash/>.

MANKU, G. S., A. JAIN AND A. D. SARMA. Detecting NearDuplicates for Web Crawling. In International World Wide Web Conference. 2007.

MÁRQUEZ, F. B. Procesamiento de Texto y Modelo Vectorial. 2013. Available from Internet:<http://www.cs.waikato.ac.nz/~fjb11/clases/irintro.pdf>.

MONGE, A.E. y ELKAN, C.P. The Field Matching Problem: Algorithms and Applications. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.*1996, pp. 267-270.

MYSQL.ORG. About MySQL. 2015. Available from Internet:<http://www.mysql.com/about/>.

NASEER, O., A. NASEER AND A. A. KHAN Using Page Size for Controlling Duplicate Query Results in Semantic Web. International Journal of Web & Semantic Technology, 2013, 4.

ORNITORRINCODIGITAL.COM. Internet en 2013: Cifras Mundiales de Actividad y Volumen. 2013. Available from Internet:<http://ornitorrincodigital.com/2013/01/17/internet-en-2013-cifras-mundiales-de-actividad-y-volumen/>.

PICONE, F. 7 sitios para detectar contenido duplicado o copypaste en Internet. 2011. Available from Internet:<http://www.dotpod.com.ar/7-sitios-para-detectar-contenido-duplicado-o-copypaste-en-internet/>.

PINGDOM, R. Internet 2012 in numbers. 2013. Available from Internet:<http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/ >.

POSTGRESQL.ORG. Sobre PostgreSQL. 2010. Available from Internet:<http://www.postgresql.org.es/sobre_postgresql>.

RIVERA ALVA, EDUARDO. 2008. Arquitectura de Software II. Diagramas de Componentes y Despliegue.[En línea] 11 de Noviembre de 2008. [Citado el: 5 de Abril de 2015.] Available from http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue.

SANTIAGO, I. Contenido Duplicado: Significado, detección y eliminación. 2013. Available from Internet:<http://ignaciosantiago.com/blog/contenido-duplicado/>.

SCHIMKE, S., C. VIELHAUER AND J. DITTMANN. Using Adapted Levenshtein Distance for On-Line Signature Authentication. In Proceedings of the 17th International Conference on Pattern Recognition. 2004.

SHAIK, N., V. VETAPALEM AND Y. RAVURI Effective Search engine optimization with Google. International Journal of Advanced Research in Computer Engineering & Technology, 2012, 1(9).

SIDOROV, G., A. GELBUKH AND H. G´OMEZ-ADORNO Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. Computación y Sistemas, 2014, 18, 491–504.

SIVAKUMAR, P. AND R. M. S. PARVATHI Sketching-Din Elimination of Web Page. Journal of Computer Science, 2011.

SLIDESHARE. Herramientas CASE. 2008. Available from Internet:<http://es.slideshare.net/guestf131a9/herramientas-case>.

SMITH, T. F. AND M. S. WATERMAN *Identification of Common Molecular Subsequences Journal of Molecular Biology*. 1981, pp. 195-197.

STEVENS, P. AND POOLEY, R. Utilización de UML. 2007.

TERESA, T. D. Por Fin, una Guía Completa para Tratar con el Contenido Duplicado. 2014, no. 1-2-3-4. Available from Internet:<http://deteresa.com/contenido-duplicado/>.

THEOBALD, M., J. SIDDHARTH AND A. PAEPCKE. SpotSigs: Robust and Efficient Near Duplicate Detection in Large Web Collections. In SIGIR'08. Singapore, 2008.

TORREGROSA, E. Los peores enemigos del SEO: El contenido duplicado (II). 2012. Available from Internet:<http://www.webpositer.com/los-peores-enemigos-del-seo-el-contenido-duplicado-ii.html>.

UCI. Análisis y Diseño con el Diagrama de Clase. In., 2015.

UHL, A. AND P. WILD. Enhancing Iris Matching Using Levenshtein Distance with Alignment Constraints. In Proceedings of the 6th International Symposium on Advances in Visual Computing. 2010, vol. 6453 p. 469-479.

VALERO, M. El plagio de tus contenidos en otras webs, herramientas y soluciones. 2011. Available from Internet:<http://www.miguel-valero.es/seo-adictos-24h/28-07-2011/el-plagio-de-tus-contenidos-en-otras-webs-herramientas-y-soluciones/>.

VALERO, M. Contenidos duplicados en tu web. Detección y corrección del problema. 2013. Available from Internet:<http://www.miguel-valero.es/seo-adictos-24h/12-07-2011/contenidos-duplicados-en-tu-web-deteccion-y-correcion-del-problema/>.

VANDYK, J. An Introduction to Drupal Architecture. In DrupalCamp Des Moines. Iowa, 2011.

WIKIPEDIA.ORG. Distancia de Levenshtein. 2015. Available from Internet:<http://es.wikipedia.org/wiki/Distancia_de_Levenshtein>.

WIKIPEDIA.ORG. Smith–Waterman algorithm. 2015. Available from Internet:<http://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm>.

WILLIAMS, K. AND C. L. GILES. Near Duplicate Detection in an Academic Digital Library. In DocEng'13. Florence, Italy, 2013.

XIAO, C., W. WANG AND X. LIN. Efficient Similarity Joins for Near Duplicate Detection. In International World Wide Web Conference. 2008.

YANCEY, W.E. Evaluating String Comparator Performance for Record Linkage. (Statistics #2005-05). Washington, DC: U.S. Census Bureau, Statistical Research Division. 2006.

YANG, M. Introduction to OpenUP. 2011. Available from Internet:<http://epf.eclipse.org/wikis/openup/>.

WEBSEGURIFY. Innovative Web Application Security Tools Available from http://www.websecurify.com/.

Glosario de términos

API: Acrónimo del inglés *Application Programming Interface*, traducido al español como Interfaz de programación de aplicaciones.

RSS: Corresponde a *Rich Site Summary* o *Really Simple Syndication*, diseñado para la distribución de noticias o información tipo noticias, contenidas en portales web y *weblogs*.

SSL: Protocolo *Secure Sockets Layer*, permite conectarse de forma segura entre dos extremos de la red, mediante técnicas de encriptación y criptografía.

Función *hash*: método para generar claves o llaves que representen de manera casi unívoca a un documento o conjunto de datos.

Anexo 1. Modelo de dominio

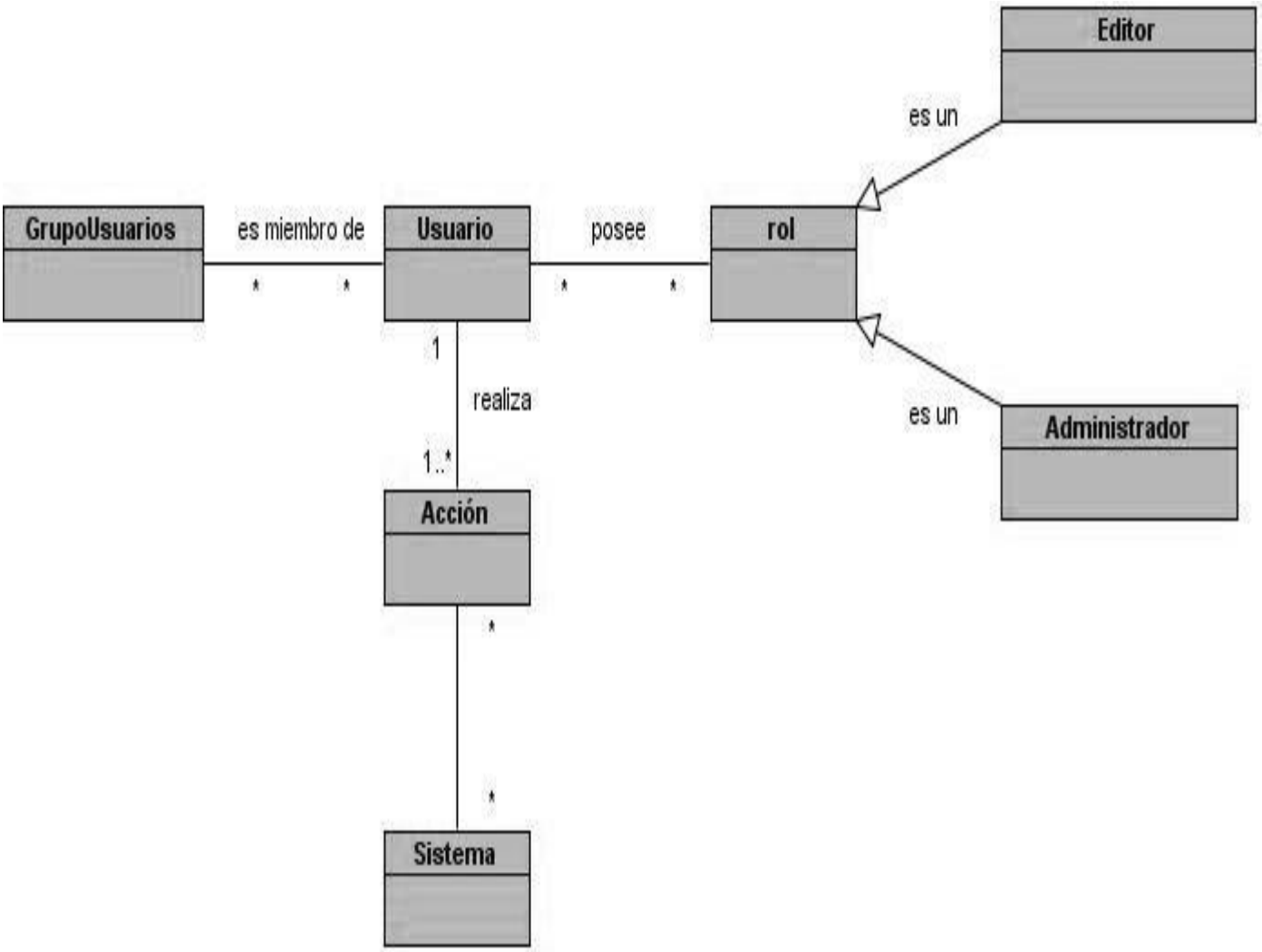

Figura 11. Modelo de dominio

Anexo 2. Descripción de casos de uso

Tabla 6. Descripción de CU Realizar búsqueda de duplicado web por metadato específico

Objetivo	El módulo realiza una búsqueda de contenido duplicado en el portal web a través de un metadato.	
Actores	administrador, editor	
Resumen		
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	**Sistema**
1.	El usuario accede al menú Reportes/ detector de contenido.	El sistema realiza una búsqueda de contenido duplicado en el portal web.

		a) Si está seleccionado el metadato URL ir a sección “Reporte URL” b) Si está seleccionado el metadato título ir a sección “Reporte título” c) Si está seleccionado el metadato cuerpo de página ir a sección “Reporte cuerpo de página” Si está seleccionado la opción título y cuerpo de página ir a sección “Reporte título y cuerpo de página”
Sección: “Reporte URL”		
1.		El sistema muestra una notificación sobre la existencia de contenido duplicado a través del metadato URL y recomienda una acción para su tratamiento.
2.	Termina el caso de uso	
Sección: “Reporte título”		
1.		El sistema muestra un reporte de los nodos que presenten contenido duplicado a través del metadato título y el porciento de similitud.

2.	Termina el caso de uso	
Sección: “Reporte cuerpo de página”		
1.		El sistema muestra un reporte de los nodos que presenten contenido duplicado a través del metadato cuerpo de página y el porciento de similitud.
2.	Termina el caso de uso	
Sección: “Reporte título y cuerpo de página”		
1.		El sistema muestra un reporte de los nodos que presenten contenido duplicado a través de los metadatos título y cuerpo de página y el porciento de similitud.
2.	Termina el caso de uso	

Relaciones	**CU incluidos**	
	CU extendidos	

Requisitos no funcionales	
Asuntos pendientes	

Anexo 3. Diagrama de clases del diseño

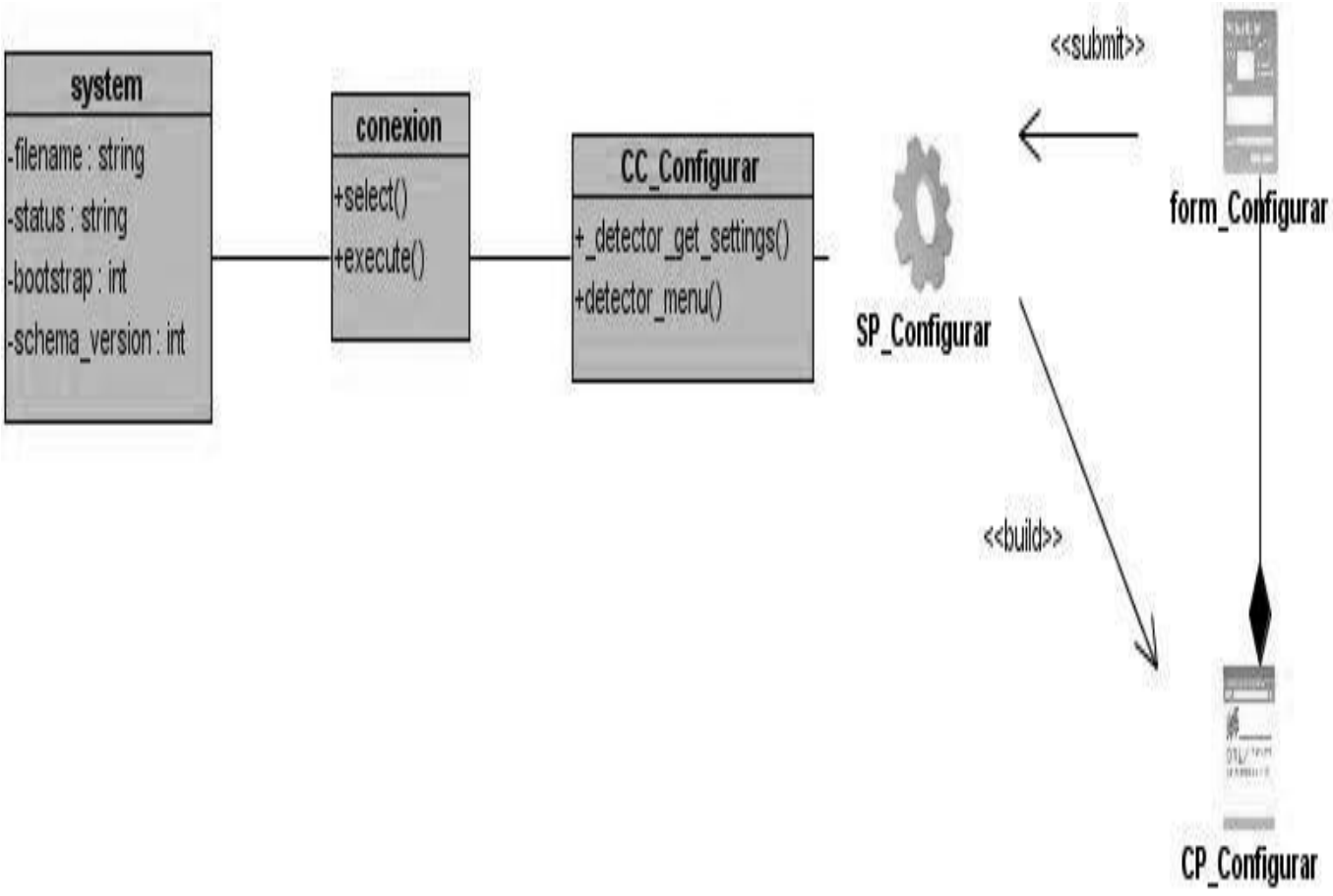

Figura 12. DCD Configurar parámetros de búsqueda

Anexo 4. Diagramas de secuencia

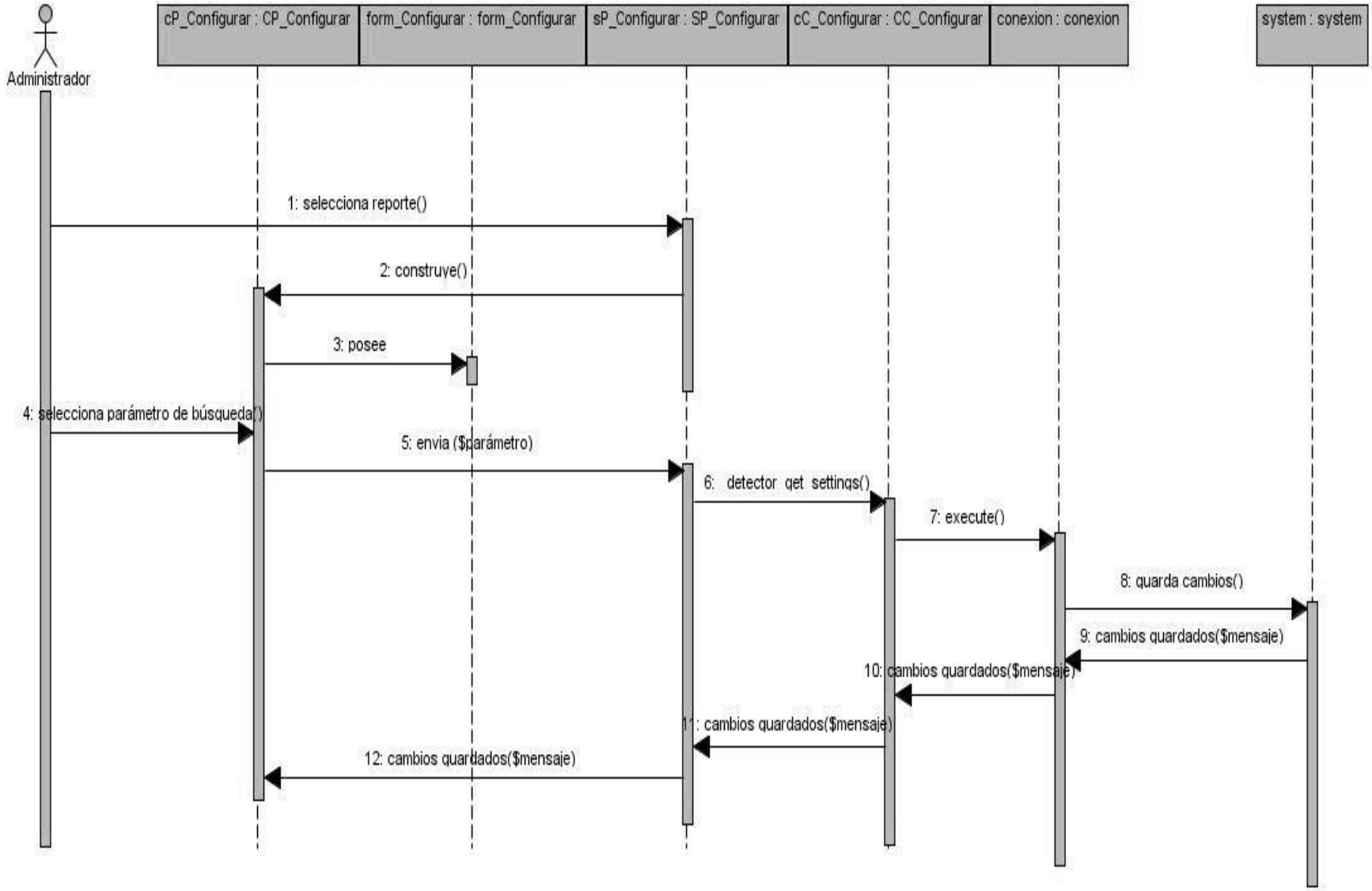

Figura 13. DS Configurar parámetros de búsqueda

Anexo 5: Casos de prueba

Tabla 7. Caso de prueba – Configurar parámetros

Escenario	**Descripción**	**Respuesta del sistema**	**Flujo central**
EC 1.1 Seleccionar metadato de búsqueda.	Permite al administrador seleccionar el metadato de búsqueda	Muestra un mensaje de configuración guardada.	1-El usuario accede al menú Reportes/ detector de contenido. 2-El sistema muestra un formulario con los metadatos a escoger. 3-El usuario selecciona un metadato. 4-El sistema guarda los cambios y muestra un mensaje de notificación.