UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1

HERRAMIENTA INFORMÁTICA PARA LA RECUPERACIÓN DE IMÁGENES DIGITALES PUBLICADAS EN LA WEB

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

Autores:

Eric Bárbaro Utrera Sust

Walter Daniel Camejo López

Tutores:

Ing. Eduardo Manuel Macías Sotolongo

Ing. Odisleysi Martínez Furones

Co-tutor:

Ing. Yulio Alemán Jiménez

La Habana, Junio 2015

"Fomentar la discusión franca y no ver en la discrepancia un problema, sino la fuente de las mejores soluciones."

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Nosotros, Eric Bárbaro Utrera Sust y Walter Daniel Camejo López, declaramos ser los únicos autores de este trabajo y autorizamos a la Facultad 1 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los____ días del mes de ____ del año ____.

Autor:
Eric Bárbaro Utrera Sust

_____

Firma del Autor

Autor:
Walter Daniel Camejo López

_____

Firma del Autor

Tutor:
Ing. Eduardo Manuel Macías Sotolongo

_____

Firma del Tutor

Tutor:
Ing. Odisleysi Martínez Furones

_____

Firma del Tutor

Co-tutor:
Ing. Yulio Aleman Jiménez

_____

Firma del Co-tutor

AGRADECIMIENTOS

A nuestras familias.

A nuestros amigos y amigas.

A nuestros profesores docentes.

A la tropa del proyecto: Eyeris, Jorgito, Paul, Adrián, Rubén, Jose, Ariagna, Yasmani y Elieyis, por estar disponible siempre para cada duda o preocupación.

A nuestros tutores: Eduardo Manuel Macías Sotolongo, Odisleysi Martínez Furones y Yulio Aleman Jiménez, por su maravillosa dedicación en todo momento al desarrollo del trabajo de diploma.

A los profesores: Serguey, Delly, Osiris, Damián y Ailyn.

A nuestros amigos: Luis, Roannel, Maikel, Arleni, Oscar, el chino y Maykel Ramírez.

A nuestros padres por su apoyo incondicional.

A nuestras parejas por escucharnos repetir la tesis en todo momento.

Los Autores.

RESUMEN

Internet es hoy uno de los principales medios de información que existe en la actualidad y cuenta con un gran volumen de contenido, entre ellos, imágenes. En ocasiones se hace necesario buscar este tipo de contenido y generalmente se recurre a la red. En los últimos tiempos se han desarrollado sistemas internacionales que permiten agilizar el proceso de búsqueda de imágenes en la red, a los que a Cuba se le dificulta el acceso debido al bloqueo económico. Con el objetivo de dar una solución a la búsqueda de imágenes en la intranet nacional, se realiza un estudio y análisis de los Sistemas de Recuperación de Información más usados a nivel mundial atendiendo a los tipos de modelos que utilizan, clasificaciones, arquitectura y principales requisitos que poseen, para posteriormente definir los factores a tener en cuenta para la solución propuesta. Con el propósito de contribuir al desarrollo de la sociedad y los servicios al ciudadano, se desarrolla una herramienta de recuperación de imágenes digitales publicadas en la Web que pretende salvar el patrimonio fotográfico en la intranet nacional. Para la implementación de la propuesta de solución, guiada por la metodología OpenUp, se seleccionaron como principales tecnologías: Nutch como mecanismo para rastrear la Web, Solr como mecanismo de indexación, Symfony como marco de trabajo PHP, Java para la implementación del plugin de Nutch que se encargará de procesar las imágenes y Visual Paradigm como herramienta para el modelado. La herramienta implementada posee un conjunto de características y funcionalidades que contribuyen a la búsqueda de imágenes a través de filtros que permiten lograr resultados más exactos disminuyendo el tiempo en hallar la información que precisan.

Palabras clave: búsqueda, herramienta, imágenes, Sistemas de Recuperación de Información, Web.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El acelerado desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) es un factor determinante en el progreso social de la humanidad. Sus contribuciones en todas las esferas de la sociedad favorecen a que cada país de una forma u otra se desarrolle y avance continuamente. A medida que se desarrollan las tecnologías, la sociedad ha tenido la necesidad de informatizar los procesos esenciales para lograr una mayor eficiencia en el trabajo cotidiano, interactuando con una gran cantidad de información digital, a través de sistemas informáticos. En este sentido, se puede decir que el desarrollo tecnológico y la digitalización de la información se han convertido en uno de los procesos más importantes del mundo moderno (Rossini, 2003).

En un mundo donde cada día son más importantes los aspectos visuales, las imágenes son utilizadas para comunicar ideas o conceptos de manera simple y concisa. En el contexto de la sociedad en red[1], surgen modos emergentes de entender y producir saberes, donde la percepción visual desempeña un papel fundamental para la construcción del conocimiento. Por su importancia para analizar, desarrollar y tomar decisiones, las imágenes digitales son esenciales para el avance de cualquier esfera, tanto en lo social como en lo político y económico. Mediante el uso de las imágenes se consigue una mayor riqueza en la expresión oral, una contextualización concreta y una descripción más detallada, en definitiva, una mayor comunicación (Sánchez, 2014).

Cuba ha estado inmersa en un profundo y novedoso proceso de transformaciones tecnológicas, a partir del cual se emprenden nuevos programas para elevar la cultura tecnológica de la sociedad. Entre los cambios más significativos se encuentra el notable crecimiento de la web cubana desde los últimos 3 años, alcanzando una cifra de más de 20 600 000 páginas web distribuidas en 6703 dominios únicos bajo .cu registrados oficialmente (Cubanic, 2014). Ver Figura 1.

[1] Sociedad cuya estructura social está construida en torno a redes de información a partir de la tecnología de información microelectrónica estructurada en Internet (Castells, 2013).

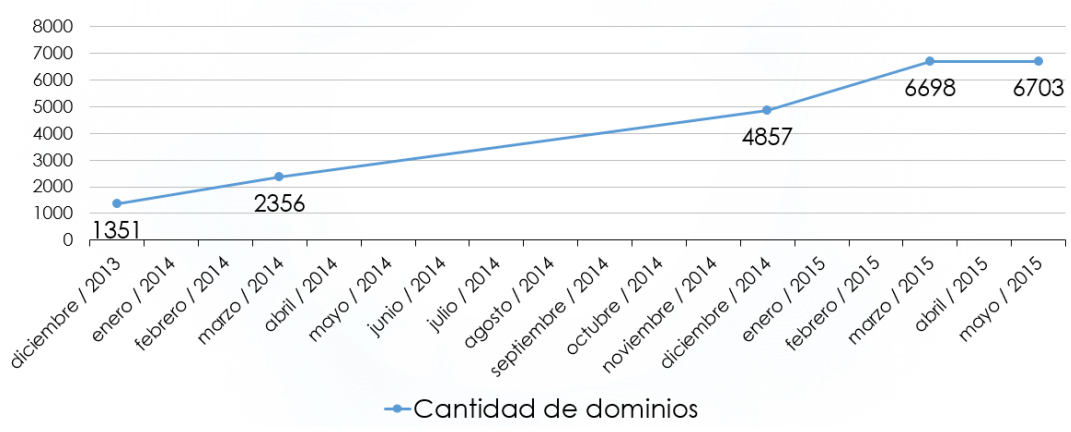


Figura 1: Historial del crecimiento de la Web cubana (2013-2015)

La búsqueda y consulta de imágenes en Cuba hasta el momento depende de los buscadores presentes en cada portal web y fundamentalmente de buscadores en Internet. Esto provoca una descentralización en la recuperación de este tipo de contenido en la intranet nacional, imposibilitando que se contribuya al acceso de este tipo de recurso por parte de los usuarios cubanos, que desde el punto de vista sociopolítico, cultural, docente, investigativo y económico, resulta de gran utilidad e interés.

Los buscadores en Internet que brindan la posibilidad de buscar imágenes digitales publicadas en la Web se apoyan en el procesamiento de éstas para obtener el color predominante de la imagen, si posee rostros, sus dimensiones, el formato, entre otras características que permiten satisfacer las necesidades de búsqueda de los usuarios. Sin embargo estos buscadores presentan un grupo de desventajas para los intereses de los usuarios cubanos, como son:

- **Manipulación de la información:** Se evidencia en la manipulación del posicionamiento de la información indexada según los intereses de sus compañías o gobiernos representativos. (Mexidor, 2011).
- **Soberanía tecnológica:** La mayoría de los buscadores en Internet obedecen las leyes del gobierno norteamericano, lo que implica que un grupo de servicios estén bloqueados, dificultando el acceso a los mismos para los usuarios cubanos (Salomón, 2014; Martínez, 2013).
- **Posicionamiento web:** Las imágenes presentes en sitios web cubanos en ocasiones no son indexadas por estos buscadores internacionales y por ende no aparecen en los resultados. Por otra parte, las imágenes que llegan a ser indexadas por lo general provienen de sitios mal

posicionados, provocando que aparezcan al final de los resultados (Carrillo y Ramírez, 2012).

En el ámbito nacional existe el buscador cubano Orión. Este sistema brinda una búsqueda básica de imágenes; pero presenta problemas en algunas métricas en relación a la recuperación de información expuestas por (Méndez, 2002):

- **Porción de la imagen indexada:** Define las porciones de la imagen que el indexador almacena, para luego comparar con las consultas de los usuarios y en base a la semejanza entre las consultas y estos fragmentos almacenados, entonces devolver las imágenes relacionadas con dicha consulta.
- **Exhaustividad:** La exhaustividad es el ratio de imágenes relevantes recuperadas en una búsqueda dada, sobre el número de imágenes relevantes para esa búsqueda contenidas en la base de datos.
- **Precisión:** La precisión es el ratio de imágenes relevantes recuperadas, sobre el total de imágenes recuperadas.
- **Tasa de fallo:** Refleja el porcentaje de imágenes recuperadas no relevantes, sobre el total de imágenes no relevantes de la base de datos.
- **Índice de irrelevancia:** Este índice es el ratio de imágenes recuperadas no relevantes a la consulta, sobre el total de imágenes contenidas en la base de datos.

El patrimonio fotográfico de la Web cubana[2], es de especial importancia y en contraposición a esto, en la actualidad no existe una forma de acceder a éste de forma efectiva. Los recursos visuales necesarios en investigaciones y otros trabajos que requieran imágenes se ven limitados por la imposibilidad de recuperar en un lugar centralizado fotografías del acervo cultural, social y nacional.

Considerando la situación problemática anteriormente descrita, se plantea como problema de la investigación: **¿Cómo contribuir a la recuperación de imágenes digitales en la intranet nacional, para facilitar el acceso al patrimonio fotográfico de la Web cubana?** Para la realización de la investigación se define como **objeto de estudio:** el proceso de recuperación de información; enfocada en el proceso de recuperación de imágenes digitales en la intranet nacional como **campo de acción**.

[2] Entiéndase en la presente investigación por patrimonio fotográfico de la Web cubana el conjunto de imágenes publicadas en portales web cubanos.

Para dar solución al problema planteado, se define como objetivo general: **Desarrollar una herramienta informática que permita recuperar las imágenes digitales publicadas en la intranet nacional, utilizando los modelos de recuperación de información y procesamiento de imágenes, para facilitar el acceso al patrimonio fotográfico de la Web cubana.**

En el marco de la presente investigación se considera que facilitar el acceso al patrimonio fotográfico de la Web cubana es brindar un lugar centralizado para la recuperación de imágenes donde se aumente la porción de la imagen indexada, la exhaustividad y la precisión, y se disminuya la tasa de fallo y el índice de irrelevancia, contribuyendo todo esto a obtener mayor calidad en los resultados de la búsqueda del usuario.

Con el propósito de cumplimentar gradualmente el objetivo general antes mencionado, el mismo se ha desglosado en los siguientes **objetivos específicos:**

1. Caracterizar los fundamentos teóricos relacionados con la recuperación de información y el procesamiento de imágenes digitales.
2. Definir las tecnologías, las herramientas y la metodología para la implementación de la herramienta informática de recuperación de imágenes digitales.
3. Diseñar la herramienta informática de recuperación de imágenes digitales.
4. Implementar la herramienta informática de recuperación de imágenes digitales.
5. Validar la herramienta propuesta.

Luego de haber tratado los elementos fundamentales del área de la ciencia a incidir y los objetivos primordiales, se formula la siguiente **hipótesis de investigación:** El desarrollo de una herramienta informática que permita recuperar las imágenes digitales publicadas en la intranet nacional, utilizando los modelos de recuperación de información y procesamiento de imágenes, facilitará el acceso al patrimonio fotográfico de la Web cubana. Teniendo en cuenta la hipótesis planteada se define como **variable independiente:** la herramienta informática para la recuperación de imágenes digitales. Esta consiste en una aplicación informática que brinda un conjunto de funcionalidades para realizar búsquedas de imágenes de forma simple y a través de diferentes filtros que permitirán encontrar resultados más exactos. Como **variable dependiente** se especifica: facilitar el acceso al patrimonio fotográfico de la Web cubana. Dicha variable hace alusión a la búsqueda de imágenes en la intranet nacional por medio de herramientas especializadas teniendo en cuenta el uso de frases y palabras claves para buscar.

Para dar cumplimiento a los objetivos específicos se definieron las siguientes tareas de investigación:

1. Valoración sobre las tendencias en el desarrollo de sistemas de recuperación y procesamiento de imágenes digitales.
2. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
3. Selección de la metodología de desarrollo de software para guiar el proceso de desarrollo de la propuesta de solución.
4. Elaboración de los artefactos requeridos por la metodología de desarrollo seleccionada.
5. Implementación de la propuesta de solución.
6. Validación de la propuesta de solución.
7. Validación de la hipótesis de la investigación.
8. Documentación de las pruebas realizadas.

En el desarrollo de la investigación se utilizaron los siguientes métodos de investigación:

Métodos teóricos

- **Histórico – Lógico:** con el objetivo de constatar teóricamente cómo ha evolucionado en el tiempo el proceso de recuperación de imágenes digitales, así como los Sistemas de Procesamiento de Imágenes Digitales (SPID), y de igual forma las herramientas y tecnologías utilizadas en el desarrollo de aplicaciones web.
- **Analítico – Sintético:** empleado para el análisis de los elementos esenciales referentes a las teorías, documentos y literatura en general relacionada con los Sistemas de Recuperación de Información (SRI) y los SPID.

Métodos empíricos

- **Modelación:** utilizado en la representación, mediante el uso de diagramas, de las características del sistema, relaciones entre objetos; y las actividades que intervienen en los procesos implementados por la herramienta de recuperación de imágenes.

Estructura del documento

Capítulo 1. Fundamentación teórica de los Sistemas de Recuperación de Información, el procesamiento de imágenes digitales y las aplicaciones web: se expondrán un conjunto de conceptos

y criterios fundamentales asociados al objeto de estudio de la investigación. Además, se estudiarán los principales sistemas de recuperación de información en la Web y disímiles técnicas de procesamiento de imágenes, con la finalidad de brindar una solución a la problemática planteada. Finalmente, se expondrán las distintas tecnologías a utilizar en el desarrollo de la herramienta, así como la metodología de desarrollo de software a utilizar.

Capítulo 2. Análisis y diseño de la herramienta de recuperación de imágenes digitales publicadas en la Web: se exponen las características del sistema, incluyendo los requisitos funcionales y no funcionales, patrones de diseño y arquitectura utilizados; además algunos de los artefactos que requiere la metodología de desarrollo utilizada.

Capítulo 3. Implementación y validación de la herramienta de recuperación de imágenes digitales publicadas en la Web: se exponen algunos aspectos asociados con la implementación de la solución informática, así como los componentes que la integran. Además, se presentan los diseños de casos de prueba a utilizar en la validación del sistema y se analizan los resultados de las pruebas realizadas que permiten evaluar la calidad de la propuesta de solución.

CAPÍTULO: 1. FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN, EL PROCESAMIENTO DE IMÁGENES DIGITALES Y LAS APLICACIONES WEB

1.1 Introducción

En este capítulo con el objetivo de favorecer una mejor comprensión de la investigación que se presenta, se realiza una revisión bibliográfica acerca de los elementos y áreas del conocimiento que engloban al objeto de estudio y al campo de acción. Además se hace un análisis y evaluación de los principales SRI y se exponen las características esenciales de las herramientas, las tecnologías y la metodología de desarrollo de software que se utilizarán para la implementación de la solución.

1.2 Fundamentos teóricos asociados al dominio del problema

A lo largo de la existencia del hombre, la información ha adquirido diversos matices en diferentes campos del saber y en distintas actividades sociales. La información como punto de partida y parte de cada proceso universal constituye hoy la base del desarrollo de las esferas de una sociedad. Según la Real Academia Española la información es: "*Comunicación o adquisición de conocimientos que permiten ampliar o precisar los que se posee sobre una materia determinada"* (Real Academia Española, 2014)*.*

Para un correcto uso de la información esta debe gestionarse de manera efectiva. La gestión de la información es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma (Dans, 2006).

1.2.1 Recuperación de información

El tamaño de la Web es imposible de medir exactamente y muy difícil de estimar. Sin embargo, se calcula que son decenas de exabytes de información, y crece permanentemente. Está formada por documentos de diferente naturaleza y formato, desde páginas HTML hasta archivos de imágenes pasando por gran cantidad de formatos estándar y propietarios, no solamente con contenido textual, sino también con contenido multimedia (Baeza-Yates y Ribeiro-Neto, 1999).

Actualmente miles de millones de usuarios emplean habitualmente la Web como medio para acceder y consultar aquella información que precisan. Ante semejante volumen de información, existen mecanismos

imprescindibles que utilizan técnicas de recuperación de información para discernir las páginas que incluyen la información buscada frente a las millones de páginas restantes que resultan irrelevantes.

Se entiende por recuperación de información el proceso por el cual, una vez analizado un documento[3], e identificada la necesidad de información, se produce una comparación entre ambos para obtener resultados satisfactorios para el usuario (Díaz, 2002).

La Recuperación de Información (RI) no es un área nueva, sino que se viene desarrollando desde finales de la década de 1950. Sin embargo, en la actualidad adquiere un rol más importante debido al valor que tiene la información y el auge y popularización de la World Wide Web. Se puede plantear que disponer o no de la información justa en tiempo y forma puede resultar en el éxito o fracaso de una operación (Baeza-Yates y Ribeiro-Neto, 1999).

También, se define la RI como "la localización y presentación a un usuario de información relevante a una necesidad de información expresada como una pregunta" (Korfhage, 1997).

Durante el proceso de Recuperación de Información para poder responder a las consultas realizadas por los usuarios se necesitan diferenciar todos los documentos relevantes y al mismo tiempo rechazar todos los documentos irrelevantes. Para ello se utilizan modelos o técnicas que ayudan a realizar este proceso de recuperación, de las cuales los más utilizados son los modelos: booleano, probabilístico y vectorial, aunque existen otras técnicas basadas en inteligencia artificial, entre las que podemos citar las redes neuronales (RN), los algoritmos genéticos (AG) o el procesamiento del lenguaje natural (PLN). (Ellis, 1996)

Modelo booleano

Constituye el primer modelo teórico, el más antiguo, empleado para establecer el subconjunto de documentos relevantes, con relación a una consulta específica. Muchos motores de búsqueda en la Web se basan en este modelo, por ser de desarrollo sencillo ya que solamente involucra el empleo de un fichero inverso[4] y una interfaz de consulta que permita computar consultas expresadas mediante palabras o expresiones booleanas.

[3] Recursos publicados en la Web (páginas web, imágenes, videos, documentos ofimáticos).
[4] Fichero donde se almacenan las palabras extraídas del texto luego de haber sido ordenadas alfabéticamente.

Un SRI booleano puro divide en dos categorías los posibles resultados de una búsqueda efectuada: satisface o no satisface. Es decir, la idea principal de este modelo es que una palabra clave puede estar ausente o presente en un documento y por tanto serán relevantes solo aquellos documentos que contengan las palabras clave especificadas en la consulta. Al considerar presente o ausente las palabras clave, los pesos de estas en los documentos serán de (0 y 1) y formarán el conjunto de documentos recuperados aquellos que tengan un valor igual a 1. Una de las principales desventajas que presenta este modelo es que no se devolverán documentos que podrían ser relevantes a pesar de que no coincidan exactamente con la consulta, pues todos ellos cumplen la fórmula en idénticas condiciones. Esta desventaja suele denominarse equiparación exacta (Baeza-Yates y Ribeiro-Neto, 1999).

Modelo probabilístico

El modelo probabilístico está compuesto por conjuntos de variables, operaciones con probabilidades y el Teorema de Bayes[5]. Está basado en el llamado "Principio de la ordenación por probabilidad". Este principio, formulado por Robertson[6], asegura que el rendimiento óptimo de la recuperación se consigue ordenando los documentos según sus probabilidades de ser juzgados relevantes con respecto a una consulta, siendo estas probabilidades calculadas de la forma más precisa posible a partir de la información disponible.

El modelo probabilístico es capaz de calcular el grado de similitud existente entre cada documento de la colección y la consulta ponderada, consiguiendo ordenar los documentos de la colección en orden descendente de probabilidad de relevancia en relación a la consulta. Es decir, éste modelo sugiere que cuantas más pruebas o evidencias se tengan sobre la consulta, sobre los documentos y sobre las relaciones entre ellos, mayores serán las probabilidades de que los resultados se adecuen a la necesidad informativa del usuario. Todo este proceso se denomina equiparación probabilística, lo que permite ordenar los documentos de los resultados conforme a su probabilidad de relevancia (Comeche, 2014). De esta manera, el modelo probabilístico supera el gran inconveniente puesto de manifiesto en el modelo

[5] **El teorema de Bayes**, enunciado por Thomas Bayes, en la teoría de la probabilidad, es el resultado que da la distribución de probabilidad condicional de una variable aleatoria A dada B en términos de la distribución de probabilidad condicional de la variable B dada A y la distribución de probabilidad marginal de sólo A. Es válido en todas las aplicaciones de la teoría de la probabilidad.

[6] **Stephen E. Robertson** es un investigador de Microsoft Research Laboratory en Cambridge, Reino Unido. Mantiene un profesorado a tiempo parcial en el Departamento de Ciencias de la Información.

booleano, a saber, la equiparación exacta.

Modelo vectorial

La idea básica de este modelo de recuperación vectorial reside en la construcción de una matriz (podría llamarse tabla) de términos o alfabeto inicial y documentos, donde las filas corresponden a estos últimos y las columnas corresponden a los términos incluidos en el alfabeto. El alfabeto se obtiene inicialmente extrayendo todos los términos de la colección de documentos y luego aplicando algoritmos de normalización, extracción de palabras poco relevantes y sin valor; así como eliminando las repeticiones de los mismos términos entre otras técnicas utilizadas. De esta forma, las columnas de esta matriz estarían representadas por una palabra o término determinado del alfabeto y las filas (que en términos algebraicos se denominan vectores) serían equivalentes a los documentos que se expresarían en función de la frecuencia de cada término.

Tanto los documentos como las consultas pueden tratarse matemáticamente como vectores en un espacio **n** dimensional, del cual viene dado el nombre de modelo vectorial. Este modelo permite además, una vez calculada la similitud entre cada documento de la colección y la consulta, ordenar todos los documentos de la colección en orden decreciente de su grado de similitud con la consulta, incorporando de este modo a los resultados aquellos documentos que satisfacen solo parcialmente los términos de la consulta. Se efectúa, en consecuencia, equiparación parcial (Comeche, 2014). El modelo booleano tiene la desventaja de no saber la equiparación exacta y el modelo probabilístico no usa frecuencias de términos dentro del documento ni longitud de documento, problemas que el modelo vectorial resuelve, lo que lo hace ser el modelo más usado en la actualidad.

Teniendo en cuenta lo antes expuesto, los presentes autores consideran que la Recuperación de Información es la disciplina que permite organizar grandes cúmulos de información, utilizando diversas técnicas y modelos matemáticos; de manera que pueda ofrecerse a los usuarios por medio de consultas, para satisfacer sus necesidades de búsqueda.

1.2.2 Sistemas de recuperación de información

Los Sistemas de Recuperación de Información (SRI) son los encargados de buscar información dentro de un grupo de documentos que han sido indizados previamente. Estos sistemas no se limitan a buscar dentro de documentos solamente, sino que pueden ampliar su dominio de búsqueda a bases de datos y también a Repositorios Institucionales (Méndez, 2004). Es importante señalar que estos sistemas pueden

obtener información a partir de búsquedas dentro de archivos multimedia como pueden ser videos, música e imágenes. El objetivo fundamental de este tipo de sistemas es brindar a los usuarios un conjunto de documentos ordenados por relevancia.

Generalmente, los sistemas de recuperación de información comparten una misma arquitectura base (Herrera, 2006), la cual se detalla a continuación:

Interfaz: un usuario con necesidades de información bien definidas, interactúa con la interfaz del sistema, mediante la cual introduce las consultas al mismo. La interfaz puede estar basada en una página web, una interfaz de escritorio o ambas.

Sistema de Formulación de Consultas: realiza un preprocesamiento de las consultas transformando las consultas hechas en el lenguaje natural a consultas entendibles por los sistemas de información.

Mecanismo de evaluación de consultas: compara los documentos representados en el sistema de información con la consulta previamente procesada, para obtener un subconjunto de documentos ordenados de acuerdo a un criterio de relevancia, y que satisfagan la consulta realizada por el usuario (Herrera, 2006).

Mecanismo de rastreo: componente que se encarga de rastrear la web siguiendo la estructura hipertextual de la misma para almacenarlos en un lugar para su posterior análisis, en muchas ocasiones es llamado también araña o araña web.

Los SRI, teniendo en cuenta la forma de operar, el alcance que poseen, los tipos de documentos que recuperan, se clasifican principalmente en directorios, metabuscadores y buscadores, constituyendo esta última la más idónea para la implementación de sistemas autónomos dedicados a la recopilación, procesamiento y recuperación de grandes volúmenes de información.

1.2.3 Buscadores

Los buscadores o motores de búsqueda son Sistemas de Recuperación de Información que permiten obtener aquellos documentos de mayor relevancia, a partir de un criterio de búsqueda introducido por el usuario. Desde la perspectiva del usuario, los buscadores deben cumplir dos requisitos fundamentales: “un tiempo corto de respuesta y una gran colección de documentos web disponibles en su índice. La calidad de un buscador reside en lo abundante, relevante y actualizada que sea su colección” (Baeza-Yates y Ribeiro-Neto, 1999).

Como muestra la Figura 2, un buscador está compuesto por tres mecanismos elementales: una interfaz de usuario mediante la cual se hacen las búsquedas y se muestran los resultados, un mecanismo de indexación el cual almacena todos los documentos que han sido recuperados y un mecanismo de rastreo que es el encargado de rastrear la Web en busca de información.

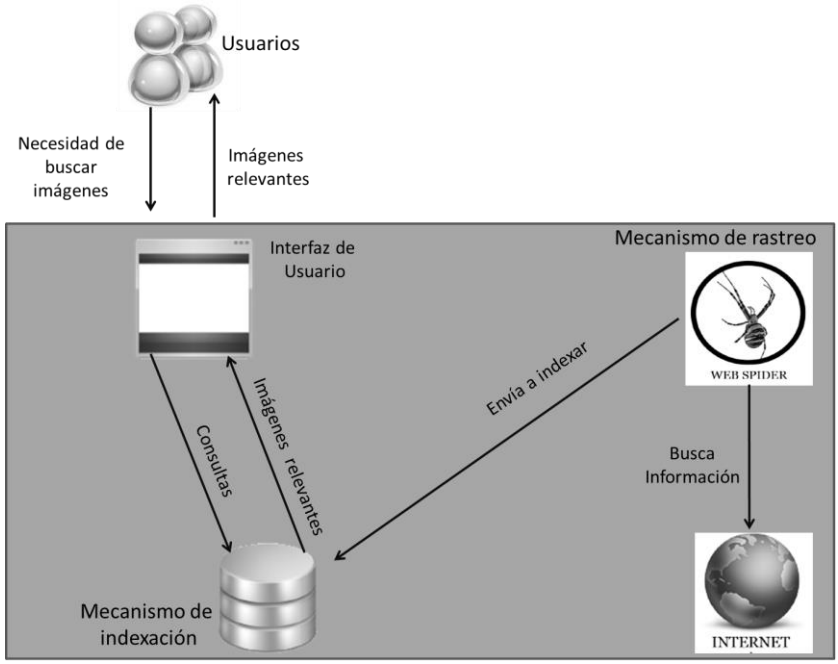


Figura 2: Arquitectura de un Buscador.
Fuente: (Herrera, 2006).

Teniendo en cuenta el objetivo general de la investigación, se hace necesario el estudio del concepto imagen digital y de su procesamiento.

1.2.4 Imagen digital

Entiéndase como imagen digital aquella que es capturada mediante un equipo electrónico y representada como un archivo de información, el cual es leído como una serie de impulsos eléctricos (De la Fraga, 2001). Las imágenes digitales son de gran importancia en el mundo actual, por lo que su procesamiento es fundamental para la tecnología y las ciencias actuales. El procesamiento de imágenes digitales es utilizado hoy en día en campos como medicina, astronomía, ingeniería, computación e industria, para extraer características presentes en una imagen como pueden ser tamaño, color predominante,

dimensiones, tipo de imagen, objetos presentes en la imagen, entre otras.

1.2.5 Procesamiento de imágenes

El término procesamiento de imágenes digitales (PID) se refiere a la manipulación y análisis de imágenes por computadora. Se utiliza particularmente para revelar información sobre imágenes, lo que involucra hardware, software y soporte teórico (De la Fraga, 2001). Para el tratamiento de una imagen digital se emplean diversas técnicas y algoritmos que tienen en su trasfondo la aplicación de múltiples ecuaciones matemáticas. Por lo general las etapas que se aplican para el procesamiento digital de imágenes son: **captura, pre-procesamiento, segmentación, extracción de características e identificación de objetos** (Wainschernker y otros, 2011).

En esta investigación solo se abordará acerca de las tres últimas etapas: segmentación, extracción de características e identificación de objetos. Las etapas iniciales no serán abordadas debido a que en la presente investigación no se realiza la captura de imágenes con equipos electrónicos ni se modifican las imágenes para mejorar la calidad o resaltar los detalles que interesan.

En el contexto de la presente investigación se utilizarán las tres últimas fases para extraer los metadatos[7] presentes en una imagen. La fase de segmentación se utilizará para dividir la imagen en segmentos y a partir de la imagen segmentada poder identificar si la misma posee desnudos. En la fase de extracción de características se obtendrán metadatos de la imagen como pueden ser: color predominante, tamaño, si posee fondo transparente, dimensiones en centímetros y en píxeles[8], la disposición de la imagen, entre otras características. Por último, la etapa de identificación de objeto permitirá identificar rostros presentes en una imagen.

1.2.6 Estándar de representación de metadatos

En la actualidad la interoperabilidad entre las aplicaciones informáticas es importante ya que es la capacidad que tiene un producto o un sistema para funcionar con otros productos, sistemas existentes o futuros sistemas a desarrollarse (Gill y otros, 2008). Teniendo en cuenta lo anterior, la utilización de un estándar de representación de metadatos posibilitará la interoperabilidad de la herramienta de

[7] Los metadatos son datos estructurados que describen información, describen el contenido, la calidad, la condición y otras características de los datos (Gill y otros, 2008).
[8] Los píxeles son diminutos puntos de color que forman una imagen digital.

recuperación de imágenes digitales publicadas en la Web con otros sistemas. En el marco de la presente investigación se estudiaron los siguientes estándares de representación de metadatos para imágenes:

EXIF

Exif es una abreviación de “Exchangeable Image File Format” que se traduce como “Formato de intercambio de imagen “o “Formato de fichero de imagen intercambiable”. El formato EXIF fue creado por la Asociación de Desarrollo de la Industria de Japón como un formato de fichero de imagen para cámaras digitales facilitando la inclusión en los ficheros de imagen de datos sobre la misma (autor, fecha de captura, espacio de color, pixelXdimension, pixelYdimension, entre otros (Lapuente, 2014).

IPTC

Tradicionalmente conocidos por "encabezados IPTC" estos metadatos han sido desarrollados por el IPTC (International Press Telecommunications Council). Está pensado para facilitar el intercambio de información entre agencias de noticias y ofrece metadatos sobre autor, título, descripción, palabras clave, urgencia, derechos, entre otros (Lapuente, 2014).

XMP

Es un estándar originalmente creado por Adobe Systems Inc., para la creación, procesamiento e intercambio de metadatos normalizados y las costumbres de los documentos digitales. También proporciona directrices para la incorporación de la información XMP en formatos de imagen, vídeo y archivos de documentos populares, como JPEG y PDF, sin romper su legibilidad por las aplicaciones que no admiten XMP (Lapuente, 2014).

Se seleccionó como estándar de representación de metadatos EXIF por ser usado por la mayoría de las cámaras digitales actuales y además porque los metadatos generados por este estándar se corresponden en gran parte con los metadatos que serán identificados a partir del procesamiento de imágenes.

1.3 Estudio de herramientas existentes para la recuperación de imágenes digitales en la Web

En la actualidad existen en el mundo diversas aplicaciones web que permiten realizar búsquedas avanzadas de imágenes, atendiendo a diferentes criterios seleccionados o introducidos por un usuario. Algunas de las más utilizadas son Google, Bing y Yahoo (NETMARKETSHARE, 2014), ver Anexo 1.

Google

Google Inc. es una empresa multinacional estadounidense especializada en productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías (Google Inc., 2013). Google brinda la posibilidad de buscar imágenes a partir de diferentes criterios de búsqueda. Algunos de estos criterios son:

- **Mostrar imágenes dada una o varias palabras:** muestra imágenes asociadas a todas las palabras del criterio de búsqueda.
- **Mostrar imágenes dada una frase exacta:** muestra imágenes que contengan exactamente el criterio de búsqueda.
- **Mostrar imágenes asociadas a cualquiera de varias palabras:** agrega el operador lógico disyuntivo “OR” entre las palabras del criterio de búsqueda.
- **Mostrar imágenes que no estén asociadas a ninguna de las palabras:** agrega el operador de negación “-” delante de la palabra que no se quiere incluir.
- **Mostrar imágenes por tamaño:** muestra imágenes según el tamaño deseado.
- **Mostrar imágenes por color:** muestra imágenes donde predomine un color en particular o varios colores seleccionados previamente por el usuario.
- **Mostrar imágenes por tipo:** muestra imágenes según el tipo de imagen deseada por el usuario. Pueden ser rostros, fotografías, dibujos en líneas, animadas entre otros.

Bing

Es un buscador web de Microsoft anteriormente conocido como Live Search, Windows Live Search y MSN Search. Algunos de las opciones de búsqueda avanzada de imágenes que brinda Bing son las siguientes:

- **Búsqueda de imágenes por tamaño:** esta opción al igual que en Google muestra imágenes según el tamaño deseado ya sea mediano, pequeño, papel tapiz, grande, etc.
- **Buscar imágenes por color:** muestra imágenes donde predomine un color en particular o varios colores seleccionados previamente por el usuario.
- **Por tipo de imagen:** muestra imágenes según el tipo de imagen deseada por el usuario.
- **Buscar imágenes por diseño:** muestra imágenes cuadradas, rectangulares entre otros diseños.

Yahoo! Search

Yahoo! Search es un motor de búsqueda, propiedad de Yahoo! Inc. empresa global de medios con sede en Estados Unidos. Yahoo! Search como gran buscador también brinda una búsqueda avanzada de imágenes la cual brinda la posibilidad de buscar por:

- **Búsqueda de imágenes por tamaño:** esta opción al igual que en los demás buscadores muestra imágenes según el tamaño deseado por el usuario aunque esta se diferencia de los demás permite buscar imágenes con una proporción determinada para fondos de pantalla.
- **Buscar imágenes por color:** muestra imágenes donde predomine un color en particular o varios colores seleccionados por el usuario.

Aol.

Aol. es un buscador perteneciente a AOL Inc. anteriormente conocida como America Online, empresa de servicios de internet y medios con sede en Nueva York. Ha vendido franquicias de sus servicios a empresas en varios países alrededor del mundo o establecido versiones internacionales. Aol., al igual que Bing y Google, ofrece también criterios de búsquedas de imágenes. A continuación se describen:

- **Mostrar imágenes por tamaño:** muestra imágenes atendiendo a un tamaño determinado.
- **Mostrar imágenes por tipo:** muestra imágenes por rostro, fotografías, etc.
- **Mostrar imágenes por color:** muestra imágenes donde predomine un color en particular o varios colores seleccionados previamente por el usuario.

Orión

El Motor de Búsqueda Orión es un sistema de recuperación de información que permite realizar búsquedas sobre los contenidos publicados en la Web cubana mediante una interfaz amigable e intuitiva. Esta herramienta está compuesta por tres componentes (mecanismo de rastreo, mecanismo de indexación, interfaz web) que en su conjunto brindan las funcionalidades de búsqueda general y avanzada de documentos publicados en la Web. Orión está desarrollado con Nutch en su versión 1.5 para el rastreo en la red, Solr versión 3.6 como mecanismo de indexación, Symfony en su versión 2.0 para la interfaz web y la interacción con el mecanismo de indexación, MongoDB versión 2.0 como gestor de base de datos no relacional para almacenar estadísticas, Java versión 7.0 para el manejo de plugins en Nutch y Solr y HTML 5 y CSS 3 para la interfaz web (Hernández, 2013).

Como resultado de este estudio se pudieron distinguir las funcionalidades brindadas por algunos de los

buscadores más utilizados en el mundo con respecto a la búsqueda avanzada de imágenes. Actualmente en Cuba el buscador Orión es la herramienta de recuperación de información más completa existente en el país, este buscador aunque aún se encuentra en proceso de desarrollo ha sido desplegado en determinadas instituciones con el objetivo de realizarle pruebas funcionales. De esta forma, Orión se ha convertido hoy en la principal herramienta cubana que permite la recuperación de información existente en la intranet nacional.

Orión cuenta con una búsqueda de imágenes muy básica, por lo que no satisface la necesidad existente hoy en día con respecto al acceso al patrimonio fotográfico nacional. El tiempo en obtener la imagen requerida a partir de un criterio de búsqueda introducido por el usuario no es el mejor. Todo esto motiva a desarrollar un sistema de búsqueda avanzada de imágenes para la Web cubana que permita la recuperación y accesibilidad de las imágenes presentes en los sitios web nacionales y así potenciar el uso de las nuevas tecnologías de la información y las comunicaciones a favor del desarrollo de la economía nacional, la sociedad y el servicio al ciudadano en Cuba.

De manera general y con apoyo en la Tabla 1 se puede apreciar que un grupo de funcionalidades coinciden en los buscadores. A continuación se muestra una tabla en la que se recogen las principales opciones de búsqueda avanzada de imágenes que brindan estos buscadores y su relación con cada uno de los mismos.

Tabla 1: Comparación entre buscadores en cuanto a la búsqueda de imágenes.

Funcionalidades	**Buscadores**				
	Google	**Bing**	**Yahoo! Search**	**Aol.**	**Orión**
Búsqueda de imágenes por tamaño	Sí	Sí	Sí	Sí	No
Búsqueda de imágenes por color	Sí	Sí	Sí	Sí	No
Búsqueda de imágenes por tipo	Sí	Sí	Sí	Sí	No
Mostrar imágenes dada una o varias palabras	Sí	No	No	No	No
Mostrar imágenes dada una frase exacta	Sí	No	No	No	No
Mostrar imágenes asociadas a cualquiera de varias palabras	Sí	No	No	No	No
Mostrar imágenes que no estén asociadas a ninguna de las palabras	Sí	No	No	No	No

Buscar imágenes por dimensión dado en píxeles	Sí	No	No	No	No
Buscar imágenes de personas	Sí	Sí	No	No	No

Teniendo en cuenta que las herramientas homólogas estudiadas son privativas, que el acceso a las mismas se dificulta por el limitado acceso a internet que tiene Cuba debido al embargo económico y que hoy el buscador cubano no brinda la posibilidad de realizar una búsqueda de imágenes a partir de filtros que ayudan al usuario a obtener resultados más exactos, se evidencia la necesidad de contar con una herramienta que permita la recuperación de imágenes digitales publicadas en la Web y brinde al usuario una serie de criterios de búsqueda que ayuden a obtener resultados más exactos.

1.4 Herramientas, lenguajes y tecnologías

Para desarrollar la propuesta de solución, necesidad surgida del estudio realizado de herramientas homólogas, se hace necesario estudiar varias tecnologías, lenguajes y herramientas disponibles para llevar a cabo el objetivo general de la investigación.

1.4.1 Mecanismos existentes para la recolección de información

Recolector web

Como muestra la Figura 2, la arquitectura de un buscador está compuesto por varios subsistemas que hacen que el mismo cumpla su función. Uno de esos subsistemas es el mecanismo de rastreo o araña web, el cual como se explicó anteriormente, se encarga de rastrear la Web en busca de documentos. Un *spider o araña,* puede definirse como una aplicación que se conecta a Internet periódicamente y recorre la Web en busca de información pública (Kumar, 2010).

Los ejemplos más conocidos dentro de los recolectores web atendiendo a su uso masivo, efectividad y licencia son Wire, MnoGoSearch y Nutch (Camargo y otros, 2013). A continuación se describen una serie de características de cada uno.

Wire

Wire[9] es un proyecto iniciado por el Centro de Investigación Web de Chile dirigido por el Dr. Ricardo Baeza-Yates, para crear una aplicación que permita la recuperación de información; diseñada para ser utilizada en la Web (Department of Computer Sciences University of Chile, 2011).

[9] Sitio oficial: (http://www.cwr.cl/projects/WIRE/)

Actualmente el software WIRE incluye:

- Un formato simple para almacenar una colección de documentos web.
- Un rastreador web.
- Herramientas para la extracción de las estadísticas de la colección.
- Herramientas para la generación de informes acerca de la colección.

Las principales características de WIRE son las siguientes:

- **Escalabilidad:** diseñado para trabajar con grandes volúmenes de documentos, ha sido probado con varios millones de documentos.
- **Prestaciones:** programado en C/C++ para un alto rendimiento.
- **Configurable:** todos los parámetros para el rastreo y la indexación se pueden configurar a través de un archivo XML.
- **Análisis:** incluye varias herramientas para analizar, extraer estadísticas y la generación de informes sobre sub-conjuntos de la Web, por ejemplo: la web de un país o de una gran intranet.
- **Licencia:** el código está libremente disponible (Department of Computer Sciences University of Chile, 2011).

MnoGosearch

MnoGoSearch[10] es un motor de búsqueda de código abierto y basado en SQL. MnoGoSearch consiste en dos partes. La primera parte es un mecanismo de indización (indexer) el cual se mueve a través de vínculos de hipertexto HTML y almacena información acerca de los documentos en la base de datos. La segunda parte es una interfaz web CGI[11] la cual muestra en el navegador un formulario HTML y los resultados de búsquedas utilizan información recopilada por el indizador. Trabaja bajo licencia GNU Public licence GPL (Barkov, 2014).

Entre sus principales características destacan:

- Soporte para diversos protocolos: HTTP, HTTPS, FTP, NNTP.
- Soporte para autenticación de proxy.
- Interfaces web CGI, Perl y PHP.
- Lenguaje de consulta booleano.

[10] Sitio oficial: (http://www.mnogosearch.org/)
[11] CGI: Acrónimo de Interfaz de entrada común (Common Gateway Interface en inglés)

- Soporte para la mayoría de los conjuntos de caracteres modernos.
- Soporte para múltiples bases de datos: MySQL, PostgreSQL, SQLite, Mimer, Virtuoso, Interbase, Oracle, MS SQL, DB2, Sysbase.
- Posee una API externa para PHP.
- Manejo de clústeres de base de datos.

Nutch

Nutch[12] es un *spider* o *web crawler* libre y de código abierto, desarrollado totalmente en Java por *The Apache Software Foundation*. Inicialmente fue implementado sobre la base de Apache Lucene aunque ya la versión actual es independiente de Lucene, una librería de alto rendimiento para la búsqueda basada en texto y que utiliza una modificación del algoritmo *"Vector Space Model"* (Modelo de Espacio Vectorial en español), con un enfoque booleano que restringe las estimaciones de los resultados obtenidos (Nieto, 2009). Mantiene internamente un ranking a partir del cual se define el orden de rastreo de las urls.

Nutch, como mecanismo de rastreo, posee ciertos componentes llamados *parsers*, los cuales se encargan de descomponer las páginas web y analizar cada uno de los recursos que la componen o tienen relación. Uno de estos *parsers* se denomina Tika[13], el cual puede descomponer una gran cantidad de documentos, entre ellos HTML, documentos ofimáticos, pdf y muchos más. Actualmente, aunque Tika soporta una variedad de formatos sobre una gran cantidad de tipos de documentos, no es capaz de obtener toda la información que se pudiera obtener de las imágenes que encuentra, constituyendo una de sus debilidades (The Apache Software Foundation-Tika, 2014).

En la Tabla 2 se recogen una serie de características de estos tres mecanismos de rastreo a fin de realizar una pequeña comparación entre ellos:

Tabla 2: Comparación entre mecanismos de rastreo.

Características	**MnoGoSearch**	**Wire**	**Nutch**
Multihilo	Sí	Sí	Sí

[12] Sitio web oficial http://nutch.apache.org/

[13] Conjunto de herramientas que detectan y extraen metadatos y textos contenido en documentos usando librerías de parseo (The Apache Software Foundation-Tika, 2014).

Documentos que recopila	HTML, TXT, PDF, XML, PPT, DOC, RTF, JPG, GIF, PNG, ODT	HTML, TXT, PDF, XML, PPT, DOC, RTF	HTML, TXT, PDF, XML, PPT, DOC, RTF, JPG, GIF, PNG, ODT
Configuración	Ficheros XML	Ficheros XML	Ficheros XML
Lenguajes de programación	C++	C/C++	Java
Extensible	No	Sí	Sí
Servidor de Índices	MnoGoSearch	Switch-e	Solr
Plataforma	Windows, Unix, GNU/Linux	Windows, Unix, Mac OS, GNU/Linux	Windows, Unix, Mac OS, GNU/Linux
Licencia	GNU Public licence GPL	Apache Software Foundation(ASF)	GNU Public licence GPL

Tabla tomada de: (Ruedas y Delgado, 2012)

Atendiendo a que MnoGoSearch utiliza su propio mecanismo de indexación y que Wire está más enfocado a realizar análisis webmétricos, los autores determinaron que el spider a utilizar será Nutch por ser de propósito general, porque facilita la incorporación de nuevos plugins[14] que permiten el proceso de filtrado de imágenes digitales y además porque es el mecanismo de rastreo utilizado por el buscador cubano Orión, lo que facilitará la integración entre ambos sistemas.

1.4.2 Mecanismos existentes para la indexación de información

Otro de los subsistemas presentes en la arquitectura de un buscador es el mecanismo de indexación. Entre los más utilizados se encuentran (Slideshare, 2010):

Sphinx

Motor de búsqueda de texto completo, distribuido públicamente bajo licencia GPL, su nombre es un acrónimo que se descodifica oficialmente como SQL Phrase Index. Este ofrece la funcionalidad de búsqueda rápida y relevante de texto completo para aplicaciones clientes. Fue especialmente diseñado para integrarse con bases de datos SQL, y ser de fácil acceso para los lenguajes script. Sin embargo, Sphinx no depende ni requiere ninguna base de datos específica para funcionar (Nugraha, 2014).

Entre sus principales características se encuentran:

[14] Un plugin es complemento que necesita de una aplicación principal para funcionar el cual añade una funcionalidad adicional o una nueva característica al software sin requerir cambios en el software original.

- Posee una alta velocidad de indexación (hasta 10-15 MB / seg por núcleo);
- Posee una alta velocidad de búsqueda (hasta 150-250 consultas / seg por núcleo contra 1.000.000 documentos, 1.2 GB de datos);
- Conjunto de resultados avanzados de post-procesamiento (SELECT con expresiones, WHERE, ORDER BY, GROUP BY, HAVING entre otros, sobre los resultados de búsqueda de texto);
- Comprobada escalabilidad hasta miles de millones de documentos, terabytes de datos y miles de consultas por segundo (AKSYNOFF, 2014).

Sphinx es adecuado para aquellos que ya tienen contenido digital existente dentro de los servidores de bases de datos tales como MariaDB, PostgreSQL, MS SQL y desea que el contenido sea indexado para una rápida recuperación sin tener que convertir a otro formato primero.

Elasticsearch

Elasticsearch es un servidor de búsqueda basado en Lucene, desarrollado en Java y ha sido liberado como código abierto bajo los términos de la Licencia Apache. Proporciona un motor de búsqueda de texto completo y con capacidad multiusuario, expone su funcionalidad a través de una interfaz REST recibiendo y enviando datos en formato JSON y oculta mediante esta interfaz los detalles internos de Lucene. Esta interfaz permite que pueda ser utilizado por cualquier plataforma, o sea, no solo desde Java, además puede usarse desde Python, .NET, PHP o incluso desde un navegador con JavaScript. Es persistente y de índice incremental, es decir, que los nuevos documentos a indexar son agregados a los ya existentes en la base de datos del indexador.

Entre sus características se destacan:

- Expone un API HTTP de tipo RESTful, y usa JSON tanto para peticiones como para respuestas. También se puede operar usando la API nativa de Java;
- Está libre de esquemas de datos, en el sentido de que no necesita disponer de una definición explicita del esquema;
- Búsquedas Facetadas - Muestra contador para cada categoría en los resultados de búsqueda;
- Búsqueda Geo-espacial - Búsqueda por localización y distancia. (Buscar dentro de 5 km de la posición actual);
- Los documentos (datos) no necesariamente tienen que ser planos, permite elementos anidados;
- Replicación - El índice podría ser replicado y proporciona soporte para conmutación por error;

- Posee búsqueda distribuida, es decir, la búsqueda puede realizarse en varios fragmentos/índices y al concluir la misma los resultados serán agregados;
- Presenta indexación distribuida lo que significa que los documentos van a ser almacenados en distintos nodos.
- Arquitectura diseñada pensando siempre en la distribución para permitir escalar una solución de un nodo a cientos, ofreciendo alta disponibilidad, soportando grandes cantidades de datos y cortos tiempos de respuesta.

Actualmente Elasticsearch presenta una comunidad pequeña de colaboradores, y consecuentemente una base de usuarios pequeña (Elasticsearch, 2014).

Solr

Solr es una plataforma de búsquedas basada en Apache Lucene, que funciona como un "servidor de búsquedas". Sus principales características incluyen búsquedas de texto completo, resaltado de resultados y manejo de documentos (como Word y PDF). Solr es escalable, permitiendo realizar búsquedas distribuidas y replicación de índices (SETA, 2010). Actualmente es frecuentemente usado por importantes sitios en Internet como son: el sitio web de la Casa Blanca, utiliza Solr vía Drupal para la búsqueda con resaltado y facetado de sitios; Instagram[15], es una compañía de Facebook que utiliza Solr para potenciar su API de geo-búsqueda; Jobreez[16], es potenciado con Solr 4.0 para realizar búsquedas de ofertas de trabajo a través de más de 25000 fuentes, entre otros (Solr-WIKI, 2014).

Está escrito en Java, pero puede ser usado en cualquier lenguaje, simplemente usando las peticiones GET para realizar búsquedas en el índice, y POST para agregar y actualizar documentos. Fácil de configurar y usar. Una de las características principales de Solr es su API[17] estilo REST[18] (The Apache Software Foundation-Solr, 2014). Otras de sus características son:

- Ofrece un API REST y un API de Java;
- No se requiere un esquema y tipo de documento;
- Búsquedas Facetadas - Muestra contador para cada categoría en los resultados de búsqueda;

[15] Instagram: https://instagram.com
[16] Jobreez: http://www.jobreez.com
[17] API: Acrónimo de Interfaz de Programación de Aplicaciones (Application Programming Interface en inglés)
[18] Es un tipo de arquitectura de desarrollo Web que se apoya totalmente en el estándar HTTP.

- Búsqueda Geo-espacial - Búsqueda por localización y distancia. (Buscar dentro de 5 km de la posición actual);
- Los documentos (datos) tienen que ser planos, debido a que no permite elementos anidados;
- Permite importar datos desde una base de datos;
- Replicación - El índice podría ser replicado y proporciona soporte para conmutación por error;
- Búsqueda distribuida – La búsqueda puede realizarse en varios fragmentos/índices y al concluir la misma los resultados serán agregados;
- Resaltado de los resultados de búsqueda;
- Se ha desarrollado con la capacidad de extraer el contenido del archivo desde el sistema de archivos y añadirlo como parte de índice.

Solr, al igual que Elasticsearch, facilita a los programadores desarrollar aplicaciones sofisticadas y de alto rendimiento que incluyen facetado (resultado de búsqueda ordenado en columnas con cuentas numéricas). Solr se fundamenta en otra tecnología originaria de búsqueda-Lucene, una biblioteca de Java que provee la indexación y la tecnología de búsqueda. Ambos Solr y Lucene son manejados por la Apache Software Foundation (Pugh, 2009).

Como la interfaz principal de Solr es un API REST a través del protocolo HTTP, necesita un contenedor de servlets[19]. Este servidor de búsqueda tiene varios servlets que exponen distintos servicios, como UPDATE o SELECT (Apache Software Foundation, 2014).

Además, Solr es un mecanismo de indexación que utiliza un modelo de recuperación de información híbrido ya que se basa en el modelo booleano y en el modelo vectorial para establecer el subconjunto de documentos relevantes.

En la Tabla 3 se recogen varias características de estos tres sistemas a fin de realizar una pequeña comparación entre ellos:

Tabla 3: Comparación entre mecanismos de indexación.

Nombre	**Elasticsearch**	**Solr**	**Sphinx**
Documentación técnica	www.elasticsearch.org/guide	lucene.apache.org/solr/documentation.html	sphinxsearch.com/docs

[19] Servidor web capaz de ejecutar servlets de java.

Sistema Operativo		Todos con la MV de Java	Todos con la MV de Java y contenedor de servlets	FreeBSD, Linux, NetBSD, OS X, Solaris, Windows
Popularidad	Clasificación[20]	15	12	35
	Puntuación[21]	58.92	81.88	10.03
Lenguaje de implementación		Java	Java	C++
APIs y otros métodos de acceso		API de Java, API RESTful HTTP/JSON	API de Java, API RESTful HTTP	Proprietary protocol
Extensibilidad		No	Plugins de Java	No
Triggers		Si	Si	No
Integridad en la manipulación de datos		No	Bloqueo Optimista	No
Concurrencia en la manipulación de datos		No	Si	Si
Persistencia de datos		Si	Si	Si

Fuente: (Sphinx, 2014), (Gormley y Tong, 2014), (Apache.org, 2014), (Db-engines, 2014), (Db-engines-ranking, 2014)

Sphinx fue especialmente diseñado para integrarse con los servidores de bases de datos SQL, y ser de fácil acceso para los lenguajes de script, mientras que a diferencia de este, otras soluciones de indexación de contenido digital como Solr o ElasticSearch, ofrecen sus servicios en forma de una API REST y son de propósito general sin necesidad de integrarse a una base de datos SQL, descartándose así, por el equipo de desarrollo, el uso de Sphinx como mecanismo de indexación para la herramienta de búsqueda de imágenes en la Web.

Por otra parte, aunque tanto Solr como Elasticsearch gozan de una gran popularidad, ambos son escritos en Java. Elasticsearch no presenta soporte para la manipulación de datos concurrentemente, tampoco presenta soporte para garantizar la integridad de los datos luego una transacción no atómica, mientras que Solr sí (DB-ENGINES, 2015).

[20] Representa el lugar que está en el ranking mundial de los modelos de base de datos (http://db-engines.com/en/ranking)

[21] Representa el nivel de popularidad que está en el ranking de los modelo de base de datos (http://db-engines.com/en/ranking_trend/system/Elasticsearch%3BSolr%3BSphinx), (http://db-engines.com/en/system/Elasticsearch%3BSolr%3BSphinx),

Por tanto, teniendo en cuenta la anterior comparación, añadiendo que Solr hoy es el mecanismo de indexación utilizado por el buscador cubano Orión y como se muestra en la Tabla 2, es el Servidor de índice utilizado por Nutch, el equipo de desarrollo ha determinado que como mecanismo de indexación se utilizará Solr. Es importante señalar que Solr implementa una variante del modelo TFIDF (Term Frequency Inverse Document Frequency) para determinar cuánto de relevante es un documento con respecto a la consulta del usuario, lo que se conoce como fórmula de relevancia.

1.4.3 Lenguajes de programación

A continuación se presentan los lenguajes de programación seleccionados del estudio realizado.

Del lado del servidor

Java

Java es un lenguaje de programación multiplataforma que se introdujo a finales de 1995. Al programar en Java no se parte de cero. Cualquier aplicación que se desarrolle se apoya en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java). Java incorpora en el propio lenguaje muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores. Java es un lenguaje muy completo. La compañía Sun Microsystems creadora del mismo describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico” (Jalón, y otros, 2000).

PHP

PHP es un lenguaje de alto nivel e interpretado, utilizado en su mayoría para el procesamiento dinámico de información en la Web. El mismo, cuyo significado se le confiere a *“PHP Hypertext Preprocessor”* por sus siglas en inglés, puede ser incrustado en documentos HTML, pero solo puede ejecutarse en el lado del servidor (BAKKEN, 2013). Por ser de código abierto, posee una amplia comunidad internacional que colabora en el mejoramiento del código fuente, el desarrollo y actualización de librerías para el lenguaje y la traducción de la documentación del proyecto. Corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows y Macs. La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. PHP es completamente expandible.

Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código (Mariño, 2008).

Teniendo en cuenta que PHP es un lenguaje completamente expandible, que presenta una gran variedad de módulos y es libre, incluyendo el resto de las características expuestas anteriormente, se decide seleccionarlo como lenguaje del lado del servidor a utilizar, mientras que Java se utilizará para el trabajo con Nutch, el cual es el encargado de filtrar los documentos en un buscador.

Del lado del cliente

Para el trabajo en la Web se utilizará HTML, CSS y JavaScript, todos estos integrados en el marco de trabajo Bootstrap.

Marco de trabajo Bootstrap

Bootstrap, es originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como **"**responsive design**"** o diseño adaptativo. Los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración con jQuery para ofrecer ventanas y tooltips dinámicos (Otto y Thornton, 2013). Bootstrap utiliza HTML, acrónimo de *HyperText Markup Language*, es un lenguaje de publicación especificado como un estándar por el W3C (*World Wide Web Consortium*) que permite la creación de páginas web (World Wide Web Consortium, 2014), CCS para aplicar de forma consistente diferentes estilos a los documentos creados (Lie, y Bos, 2005) y JavaScript para proporcionarle cierto dinamismo a las páginas web (Sánchez, 2003).

1.4.4 Marcos de trabajo para PHP

Otro de los componentes de un buscador es la interfaz web. Para su desarrollo existen herramientas que poseen una estructura personalizable e intercambiable para desarrollar aplicaciones web. Estas herramientas son llamadas marcos de trabajo o framework. Para desarrollar la interfaz web, a partir de la selección de PHP como lenguaje del lado del servidor a utilizar para desarrollar la interfaz web, se estudiaron los framework presentes en la Tabla 4, los cuales son los más usados en el desarrollo de

aplicaciones web con este lenguaje (Acosta y otros, 2014).

Tabla 4: Framework de PHP

PHP Framework	Arquitectura MVC	Validación incorporada	Soporte mapeador de objetos	Soporte múltiples base de datos	Almacenamiento en caché de objeto	Comunidad de código abierto.	Gran cantidad de documentación
CodeIgniter	Sí	Sí	-	Sí	Sí	Sí	Sí
Symfony	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Yii	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Zend	Sí	Sí	Sí	Sí	Sí	Sí	Sí

Teniendo en cuenta las características y beneficios expuestos anteriormente se decide seleccionar Symfony en su versión 2.3 LTS. Además, es sencillo de usar, lo suficientemente flexible como para adaptarse a los casos más complejos y puede integrarse al buscador cubano Orión que contiene una interfaz web desarrollada con este popular framework (Potencier, 2011).

1.4.5 Metodología de desarrollo

Con el objetivo de lograr una mayor organización en el proceso de desarrollo de software se seleccionan las metodologías de desarrollo, las cuales constituyen una filosofía de trabajo que proporciona una base de procesos para llevar a cabo con éxito cualquier proyecto informático. Las metodologías de desarrollo brindan soporte a la toma de decisiones en un equipo de trabajo; es decir, permiten conocer qué persona hace una determinada actividad, cuándo y cómo la debe hacer.

OpenUp

OpenUp, acrónimo de Open Unified Process, es un modelo de desarrollo de software de código abierto, creado por la fundación Eclipse como parte del Framework de Modelo de Procesos de Eclipse (Eclipse Process Framework). Diseñado para el desarrollo de proyectos con un enfoque ágil mediante un proceso iterativo e incremental, que puede ser extendido en grandes o pequeñas formas para adicionar nuevos contenidos de desarrollo o personalizar el proceso para un entorno específico. Puede ser aplicado en equipos pequeños, por lo general de tres a seis integrantes, y se organiza dentro de cuatro áreas principales de contenido: Comunicación y Colaboración, Intención, Solución, y Administración. El ciclo de

vida del proyecto se estructura en cuatro fases: Concepción, Elaboración, Construcción y Transición que podrán tener tantas iteraciones como se requiera dependiendo del grado de novedad del dominio de negocio, de la tecnología a ser utilizada, de la complejidad de la arquitectura de la solución y del tamaño del proyecto, entre otros factores (Framework Process Eclipse, 2014).

Las características del equipo de desarrollo de este trabajo propician una correcta adecuación con la metodología mencionada, dado que se está en presencia de un equipo pequeño que requiere la constante comunicación con el cliente y que deberá realizar el proceso de implementación de forma iterativa según las necesidades de trabajo y requisitos previos. Además, es necesario destacar que dicha metodología está contenida en la base tecnológica definida por el centro de desarrollo CIDI al cual pertenecen los desarrolladores. Para la obtención de algunos artefactos, se hará uso de determinadas plantillas que propone el expediente de proyecto definido por la universidad.

1.4.6 Herramientas

Para agilizar la realización de determinadas actividades del proceso de desarrollo de software se seleccionó como herramienta CASE:

Visual Paradigm

Visual Paradigm es una herramienta CASE multiplataforma, que soporta el ciclo completo de desarrollo de software: análisis, diseño, implementación y pruebas. Facilita la construcción de aplicaciones informáticas con un menor coste que destacan por su alta calidad y contribuye a mejorar la experiencia de usuario mediante el diseño de un gran número de artefactos de ingeniería de software. Permite la generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería directa e inversa, la gestión de requisitos de software y la modelación de procesos del negocio (Visual Paradigm, 2014).

Entorno de desarrollo integrado (Netbeans)

Como entorno de desarrollo integrado se utilizará Netbeans. Liberado bajo el licenciamiento dual de CDDL y GPL (versión 2), NetBeans es un potente IDE para programadores que proporciona una plataforma ideal para escribir, compilar, depurar y ejecutar programas informáticos (ORACLE, 2014). Aunque inicialmente fue ideado para Java, puede ser empleado para la codificación de aplicaciones en múltiples lenguajes de programación. Este, además de ser gratuito y sin restricciones de uso, posee versiones para los distintos

sistemas operativos del mercado, convirtiéndolo en una alternativa con grandes ventajas para los desarrolladores. La estructura modular de NetBeans le proporciona estabilidad y grandes posibilidades de ser extendido gradualmente por desarrollos comunitarios, permitiendo agregar continuamente nuevas funcionalidades. Su versatilidad lo ha convertido en el IDE por excelencia entre miles de programadores alrededor del mundo (ORACLE, 2014).

Servidor web (Apache)

Para atender las peticiones de los usuarios mediante una interfaz web se utilizará como servidor web HTTP Apache. Apache 2 es un servidor web de código abierto que implementa el protocolo HTTP 1.1, caracterizado fundamentalmente por su alto nivel de configuración, modularidad, robustez y estabilidad. Desarrollado bajo la licencia ASF por *The Apache Software Foundation*, es considerado una de las mejores y más aceptadas creaciones del mundo del software libre (Netcraft, 2015). Teniendo en cuenta las estadísticas históricas y uso diario proporcionadas por NetCraft, este servidor llegó a usarse durante el 2005 en el 70% de los sitios web en el mundo, representando su cuota máxima en el mercado hasta la actualidad. Las estadísticas presentadas demuestran que Apache 2 ha sido considerado como el servidor web HTTP por excelencia desde Julio de 1998, según las estadísticas (NetCraft, 2015), logrando que millones de servidores mundiales ratifiquen su utilización.

Jmeter

JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web. JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas (The Apache Software Foundation-JMeter, 2015).

Acunetix

Acunetix comprueba los sistemas en busca de múltiples vulnerabilidades que un atacante podría aprovechar para obtener acceso a los sistemas y datos. Acunetix puede utilizarse para realizar escaneos de vulnerabilidades en aplicaciones web y para introducir pruebas de acceso frente a los problemas identificados. La herramienta provee sugerencias para mitigar las vulnerabilidades identificadas y puede utilizarse para incrementar la seguridad de servidores web o de las aplicaciones que se analizan (Acunetix, 2015).

1.4.7 Bibliotecas disponibles para el procesamiento de imágenes

Dado que los distintos productos van ganando en complejidad cada día, se hace necesaria la búsqueda y utilización de herramientas que agilicen el desarrollo de software, como son las bibliotecas, paquetes de clases, framework, módulos, etc. Existen bibliotecas que brindan funcionalidades destinadas al procesamiento de imágenes las cuales brindan ciertas ventajas con su uso. Mediante un estudio realizado se encontraron tres bibliotecas para el lenguaje Java: JavaAdvanceImaging, Image4J, OpenCV. En el Anexo 2 se muestra un resumen a modo de comparación de estas bibliotecas teniendo en cuenta una serie de características, seleccionándose la librería OpenCV para el tratamiento de las imágenes digitales recuperadas en la Web.

1.5 Conclusiones parciales

En este capítulo se trataron los elementos teóricos que dan sustento a la propuesta de solución del problema planteado, arribando a las siguientes conclusiones:

- El estudio de los conceptos asociados a la recuperación de información y procesamiento de imágenes digitales permitió lograr un mejor entendimiento de la investigación que se realiza.
- El análisis de las funcionalidades que brindan algunos de los buscadores más utilizados del mundo permitió identificar requerimientos necesarios que ayudan al usuario a realizar búsquedas más certeras.
- La falta de un sistema que recupere las imágenes publicadas en la intranet nacional, el limitado acceso a buscadores internacionales y a los resultados que muestran, hacen necesario el desarrollo de un sistema que permita la recuperación de imágenes publicadas en la web cubana.
- La selección de la metodología, herramientas y tecnologías con soporte multiplataforma y basadas en software libre, posibilitó obtener una base tecnológica enfocada en los componentes que utilizan los sistemas de recuperación de información y de procesamiento de imágenes estudiados.

CAPÍTULO: 2. ANÁLISIS Y DISEÑO DE LA HERRAMIENTA DE RECUPERACIÓN DE IMÁGENES DIGITALES PUBLICADAS EN LA WEB

2.1 Introducción

En este capítulo se abordarán aspectos fundamentales relacionados con el diseño del sistema a desarrollar. Entre los elementos a destacar se encuentran el diagrama del modelo del dominio, mediante el cual se representan las clases conceptuales significativas del problema a resolver. Como vía para definir las futuras funcionalidades de la aplicación y qué usuarios podrán tener acceso a las mismas, se generaron los artefactos relacionados a la especificación de los requerimientos funcionales y no funcionales que deberá poseer el software; así como la especificación de los casos de uso del sistema. Como parte del diseño de la aplicación se definieron los estilos y patrones de arquitectura y diseño que se emplearán para lograr buenas prácticas de diseño y programación. A lo largo del capítulo se mostrarán los principales artefactos de ingeniería de software correspondientes a los casos de uso más críticos.

2.2 Modelo de dominio

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases de software, u objetos de software con responsabilidades, sino que modela clases conceptuales significativas en un determinado problema (Larman, 2004).

Para lograr un mejor entendimiento de la presente investigación se hace necesario describir el procedimiento de búsqueda de imágenes en la Web mediante una serie de conceptos, entidades y sus relaciones, agrupándose en un modelo del dominio con el fin de contribuir a la comprensión del contexto actual del problema.

2.2.1 Descripción de Clases del Modelo de Dominio

La modelación del dominio constituye la herramienta fundamental para garantizar la comprensión y descripción de las clases o conceptos y sus relaciones más importantes dentro del contexto del problema. A continuación se presenta la descripción de los conceptos identificados por los autores en la presente investigación.

Tabla 5: Descripción de las clases del modelo del dominio.

Conceptos	**Descripción**
Usuario	Persona que realiza las búsquedas.
Buscador	Constituye una herramienta de recuperación de información en la Web.
Rastreador	Mecanismo que se encarga de recopilar los documentos en la Web.
Servidor de índice	Mecanismo que se encarga de indexar los documentos.
Interfaz web	Constituye la vista mediante la cual se le muestran los resultados al usuario.
Intranet	Red informática donde están los documentos.
Sitio web	Colección de páginas en internet relacionadas y comunes a un dominio de internet o sub-dominio.
Documento	Recursos publicados en la Web (páginas web, imágenes, videos, documentos ofimáticos).

2.2.2 Diagrama de Clases del Modelo del Dominio

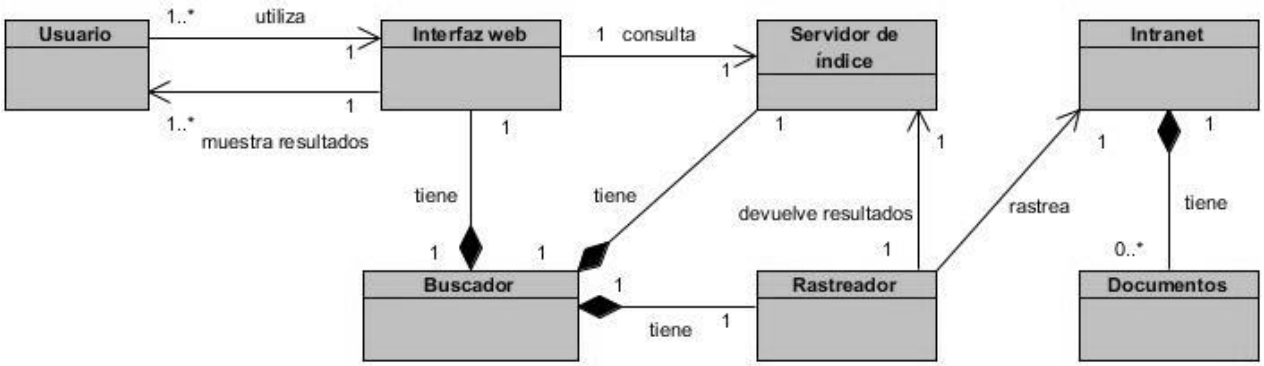


Figura 3: Diagrama de clases del modelo del dominio.

La Figura 3. muestra la relación existente entre todos los conceptos que intervienen en la investigación que se presenta. Se puede observar que intranet posee los documentos que serán primeramente recopilados por la araña y seguidamente almacenados en el componente de indexación. Tanto el indexador como la araña forman parte del buscador que contiene una interfaz web mediante la cual se le presentan los resultados al usuario.

2.3 Especificación de los Requisitos del Software

Los requisitos del software permiten establecer lo que el sistema debe hacer, sus características fundamentales, y las restricciones en el funcionamiento del sistema y los procesos de desarrollo del software. De manera general, estos requisitos expresan las necesidades objetivas que presentan los usuarios, ante un sistema que resuelve un problema en particular de un determinado dominio (Sommerville, 2005).

Luego de haber definido y modelado los conceptos asociados al dominio del problema y su relación entre cada uno de ellos, se presentan los requisitos funcionales y no funcionales de la herramienta a desarrollar.

2.3.1 Requisitos funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer (Sommerville, 2005). A continuación se presentan los requisitos funcionales de la herramienta de búsqueda avanzada de imágenes digitales publicadas en la Web.

Tabla 6: Requisitos funcionales

No.	Requisito funcional	Prioridad	Caso de uso
RF 1.	Realizar búsqueda simple de imágenes.	Alta	Buscar imagen de forma simple.
RF 2.	Realizar búsqueda de imágenes dada una o varias palabras.	Alta	Buscar por criterio.
RF 3.	Realizar búsqueda de imágenes dada una frase.	Alta	Buscar por criterio.
RF 4.	Realizar búsqueda de imágenes a partir de alguna palabra asociada al criterio de búsqueda.	Alta	Buscar por criterio.
RF 5.	Realizar búsqueda de imágenes a partir de palabras que no se encuentren asociadas al criterio de búsqueda.	Alta	Buscar por criterio.
RF 6.	Realizar búsqueda de imágenes cuadradas.	Baja	Buscar por proporción.
RF 7.	Realizar búsqueda de imágenes con disposición alta.	Baja	Buscar por proporción.
RF 8.	Realizar búsqueda de imágenes con disposición ancha.	Baja	Buscar por proporción.
RF 9.	Realizar búsqueda de imágenes dado un tamaño.	Baja	Buscar por tamaño.
RF 10.	Realizar búsqueda de imágenes por color.	Alta	Buscar por color.
RF 11.	Realizar búsqueda de imágenes con rostros.	Alta	Buscar por tipo.
RF 12.	Realizar búsqueda de imágenes por sitio o dominio.	Media	Buscar por dominio.
RF 13.	Realizar búsqueda de imágenes por extensión del archivo.	Media	Buscar por extensión.
RF 14.	Realizar búsqueda de imágenes publicadas en las	Baja	Buscar por fecha.

	últimas 12 horas.		
RF 15.	Realizar búsqueda de imágenes publicadas en las últimas 2 semanas.	Baja	Buscar por fecha.
RF 16.	Realizar búsqueda de imágenes publicadas en las últimas 24 horas.	Baja	Buscar por fecha.
RF 17.	Realizar búsqueda de imágenes publicadas en el último mes.	Baja	Buscar por fecha.
RF 18.	Realizar búsqueda de imágenes publicadas en el último año.	Baja	Buscar por fecha.
RF 19.	Realizar búsqueda de imágenes segura.	Alta	Buscar de forma segura.
RF 20.	Realizar búsqueda de imágenes dada las dimensiones en pixeles o en centímetros.	Media	Buscar por dimensiones.
RF 21.	Mostrar título de la imagen.	Alta	Mostrar información de la imagen.
RF 22.	Mostrar tamaño de la imagen.	Alta	Mostrar información de la imagen.
RF 23.	Mostrar URL[22] de la página que contiene la imagen.	Alta	Mostrar información de la imagen.
RF 24.	Mostrar dimensiones de la imagen.	Alta	Mostrar información de la imagen.
RF 25.	Mostrar nombre de la página que contiene la imagen.	Alta	Mostrar información de la imagen.
RF 26.	Mostrar tiempo de respuesta de la búsqueda.	Alta	Mostrar información de la imagen.
RF 27.	Identificar el color predominante de la imagen.	Alta	Procesar imágenes.
RF 28.	Determinar si la imagen contiene rostros.	Alta	Procesar imágenes.
RF 29.	Determinar si la imagen tiene un fondo trasparente.	Baja	Procesar imágenes.
RF 30.	Determinar tamaño de la imagen.	Baja	Procesar imágenes.
RF 31.	Determinar si la imagen tiene desnudos.	Alta	Procesar imágenes.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales, como su nombre sugiere, son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (Sommerville, 2005). A continuación se presentan los requisitos no funcionales de la herramienta de búsqueda avanzada de imágenes digitales publicadas en la web.

[22] URL: Localizador de recursos uniforme. Es una cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en la Internet.

Requerimientos de software

- RNF 1. Se requiere del sistema operativo CentOS 7.
- RNF 2. Se requiere la instalación del servidor web y de servlets Tomcat 7 para el correcto funcionamiento del servidor de Solr.
- RNF 3. Se requiere la instalación de la Máquina Virtual de Java (JVM, por sus siglas en inglés) para el correcto funcionamiento del rastreador.
- RNF 4. Se requiere la instalación del servidor web Apache en su versión 2.4 y PHP 5.5 o superior para poder visualizar la interfaz web.

Requerimientos de hardware

- RNF 5. Para el servidor de índice se necesita como mínimo: 4 GB RAM, CPU de 4 núcleos y 1GB de almacenamiento por cada 15000 imágenes aproximadamente.
- RNF 6. Para el servidor de rastreo: 4 GB RAM, CPU de 4 núcleos y al menos 80 GB de Disco Duro.
- RNF 7. Para la interfaz web: 4 GB RAM, CPU de 4 núcleos y al menos 40 GB de Disco Duro.

Requerimientos de diseño e implementación

- RNF 8. Como lenguaje de programación para la interfaz web se deberá utilizar PHP en su versión 5.5 o mayor.
- RNF 9. Como lenguaje de programación para el plugin del componente de rastreo se deberá utilizar Java.
- RNF 10. Para el desarrollo de la aplicación web se deberá utilizar Symfony 2.3 LTS o superior como marco de trabajo.

Requerimientos de apariencia o interfaz de usuario

- RNF 11. Todos los iconos irán acompañados de un texto descriptivo en las resoluciones mayores o igual a 992px de ancho.
- RNF 12. En la página de resultados siempre estará visible el menú de opciones, el formulario de búsqueda y el paginador, para resoluciones superiores a 800x600.
- RNF 13. En el formulario de búsqueda avanzada se brindará una descripción textual para cada campo.

Requerimientos de usabilidad

- RNF 14. Debe contar con la portabilidad necesaria para poder ser transferido de un ambiente a otro o reemplazado por nuevas versiones.
- RNF 15. El sistema debe permitir acceder a sus distintas partes con una profundidad máxima de 3 clics.
- RNF 16. Los dispositivos clientes que utilizarán la herramienta deben contar con navegadores web que soporten HTML5 y CSS3.
- RNF 17. Se requiere el uso de herramientas y recursos de software libre, las cuales se podrán usar, modificar y distribuir libremente.

Requerimientos de eficiencia

- RNF 18. El sistema debe ser capaz de responder como máximo en 5 segundos 5000 peticiones.

Requerimientos de seguridad

- RNF 19. Integridad: el sistema debe permitir la realización de salvas periódicas de la información.
- RNF 20. Disponibilidad: el sistema debe permitir una instalación distribuida contribuyendo al balanceo de carga y la redundancia.
- RNF 21. El sistema no debe permitir más de 20 resultados por consulta.
- RNF 22. Los formularios deben ser protegidos por un Token de seguridad.
- RNF 23. Los campos de entrada deben ser validados y escapados.
- RNF 24. Por defecto estará activada la búsqueda segura.

2.4 Modelo de Casos de Uso del Sistema

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un Caso de Uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema (Sparks, 2013).

2.4.1 Diagrama de Casos de Uso del Sistema

Para favorecer la organización y el entendimiento de los casos de uso, se muestran cada uno de estos, inicializados por el actor correspondiente, en diagramas separados. Ver Figura 4 y 5.

Figura 4: Casos de uso inicializados por el Rastreador.

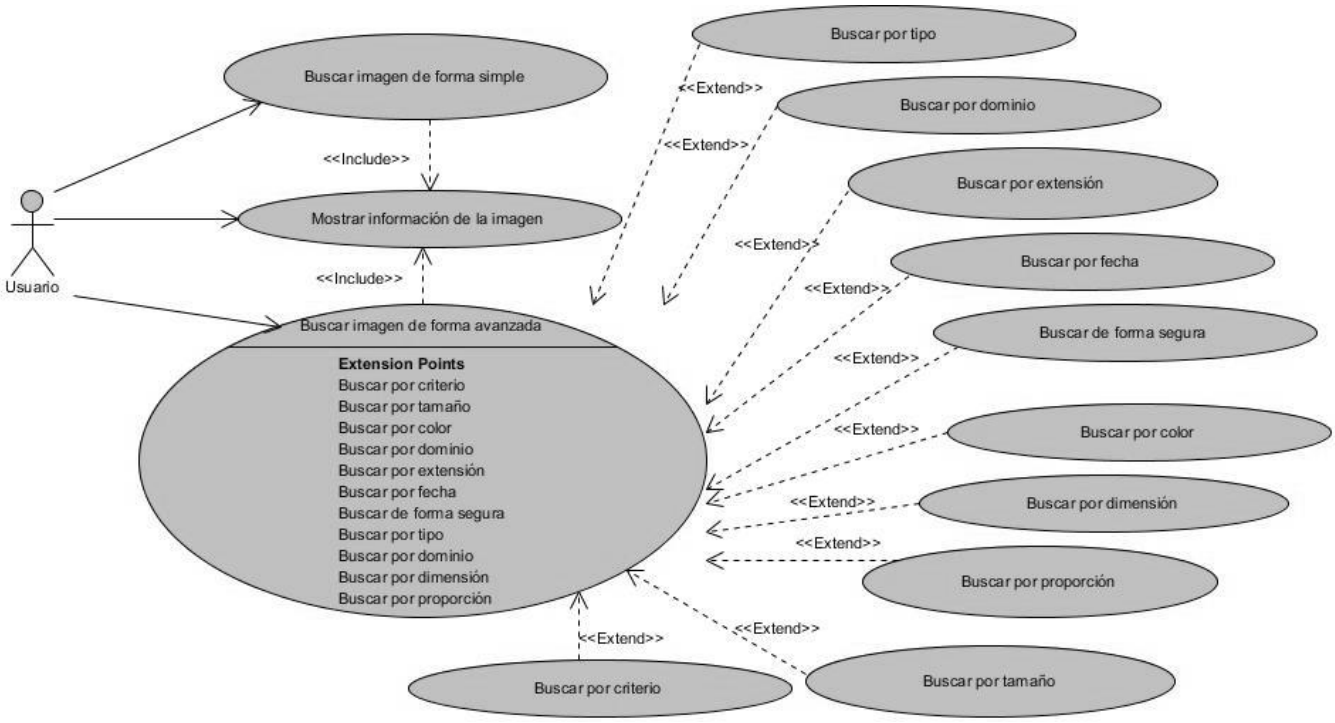


Figura 5: Casos de uso inicializados por el Usuario.

En los diagramas se encuentran representados en total 14 casos de uso y los actores que interactúa con cada uno de ellos. A continuación se presentan estos casos de uso y una breve descripción.

- **CU1 Buscar imagen de forma avanzada:** Engloba todas las formas de realizar una búsqueda avanzada por parte del usuario.
- **CU2 Buscar imagen de forma simple:** Se encarga de realizar una búsqueda de imágenes de forma sencilla y rápida.
- **CU3 Buscar por criterio:** Posee una serie de criterios para filtrar la búsqueda de imágenes. Estos criterios son: búsqueda de imágenes dada una o varias palabras, dada una frase, a partir de algu-

na palabra asociada al criterio de búsqueda y a partir de palabras que no se encuentren asociadas al criterio de búsqueda.

- **CU4 Buscar por tamaño:** Filtra la búsqueda a partir de un tamaño especificado por el usuario.
- **CU5 Buscar por color:** Realiza una búsqueda de imágenes donde predomine el color seleccionado por el usuario.
- **CU6 Buscar por tipo:** Realiza una búsqueda de imágenes por el tipo de imagen que representa. Estos tipos pueden ser: rostros y animadas
- **CU7 Buscar por dominio:** Realiza una búsqueda de imágenes por dominios de red o por sitios. Ejemplo mes.edu.cu, uci.cu, cubadebate.cu, entre otros.
- **CU8 Buscar por extensión:** Permite buscar imágenes por extensión del archivo, por ejemplo JPG, PNG, BMP, entre otros.
- **CU9 Buscar por fecha:** Filtra la búsqueda de imágenes publicadas en las últimas horas, en las últimas semanas y en las últimas 24 horas.
- **CU10 Buscar de forma segura:** Filtra imágenes que contengan desnudos y no las muestra al usuario.
- **CU11 Buscar por dimensiones:** Permite buscar imágenes a partir de las dimensiones introducidas por el usuario y el tipo de dimensión, ya sea en píxel o en centímetro.
- **CU12 Mostrar información de la imagen:** Muestra el tamaño de la imagen, sus dimensiones, el título y la URL a la que pertenece.
- **CU13 Procesar imágenes:** El rastreador extrae información de cada una de las imágenes que son recuperadas por el mecanismo de rastreo.
- **CU14 Buscar por proporción:** Filtra la búsqueda de imágenes por disposición (cuadrada, alta, ancha).

2.5 Patrones de casos de uso utilizados

Los patrones de casos de usos son comportamientos que deben existir en el sistema, ayudan a describir qué es lo que el sistema debe hacer, es decir, describen el uso del sistema y cómo este interactúa con los usuarios. Estos patrones son utilizados generalmente como plantillas que describen como debería ser estructurados y organizados los casos de uso. Son patrones que capturan mejores prácticas para modelar casos de uso (Larman, 2004). En el desarrollo de la investigación se identificaron los siguientes patrones de casos de uso:

2.5.1 Relación de Extensión

Un caso de uso extiende a otro cuando sin alterar a este, se incorpora su funcionalidad como parte integral del primero. Se denota con una relación que apunta del caso extendido al caso base y la conexión se hace o bien al principio del flujo de eventos principal del caso base o en alguno de los puntos de extensión que este haya definido. Ver Figura 6.

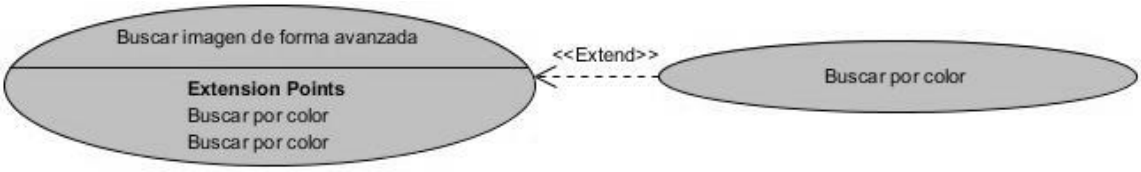


Figura 6: Ejemplo de uso del patrón “Relación extensión” en el CU inicializado por el usuario.

2.5.2 Relación de inclusión

Un caso de uso concreto incluye a un fragmento de caso de uso, cuando como parte de su descripción breve o su flujo de eventos, se hace referencia al texto del fragmento; de forma tal que lo dicho en el fragmento pasa a ser parte de la especificación del caso de uso. Ver Figura 7.

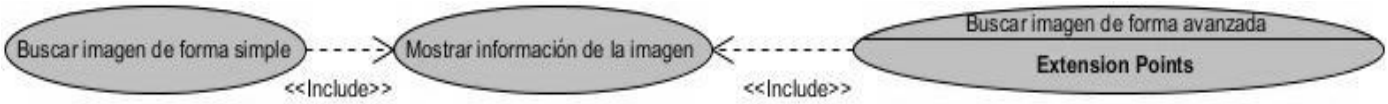


Figura 7: Ejemplo de uso del patrón “Relación de inclusión” en el CU inicializado por el usuario.

2.5.3 Especificación de casos de uso

A continuación se realiza la descripción de uno de los CU críticos de la herramienta, el resto de las descripciones se encuentran en el Anexo 3.

Tabla 7: Descripción del CU “Procesar imagen”

Objetivo	Extraer los metadatos de la imagen
Actores	Administrador (inicia), Sistema
Resumen	El sistema extrae los metadatos de una imagen previamente identificada en la red.
Complejidad	Alta
Prioridad	Alta

Precondiciones		El administrador ha configurado el sistema y ha iniciado el rastreo.
Postcondiciones		Se han extraído los metadatos de la imagen y se ha indexado la imagen en Solr con sus metadatos correspondientes.
Flujo de eventos		
Flujo básico Procesar imagen		
Actor		**Sistema**
1	Inicia el rastreo	
2		Extrae los metadatos de la imagen
3		Obtiene los datos de la imagen almacenados en la base de datos de Nutch.
4		Crea el documento a indexar en Solr.
6		Termina el CU
Relaciones	**CU incluidos**	
	CU extendidos	
Requisitos no funcionales		

2.6 Estilo arquitectónico

Un estilo arquitectónico es la organización fundamental de un sistema encarnada en sus componentes, las relaciones de los componentes con cada uno de los otros y con el entorno, y los principios que orientan su diseño y evolución (Reynoso y Kicillof, 2004).

Arquitectura del sistema

Symfony basa su funcionamiento interno en la arquitectura Modelo - Vista – Controlador (MVC), utilizada por la mayoría de frameworks web. No obstante, según su creador Fabien Potencier: "Symfony no es un framework MVC. Symfony sólo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad del usuario (Potencier, 2011). La siguiente figura muestra la arquitectura del sistema desarrollado.

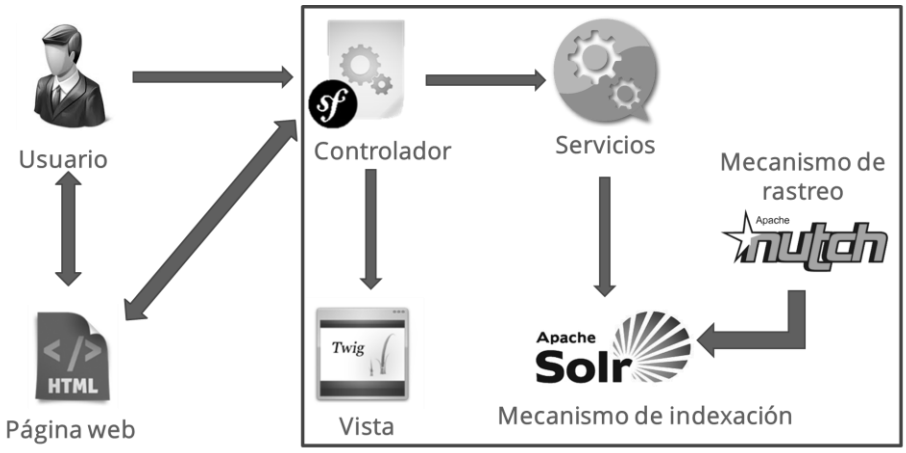


Figura 8: Arquitectura del sistema.

Como se observa en la Figura 8 a través del controlador son recibidas y atendidas todas las peticiones al sistema. Cuando el controlador recibe una petición del usuario, consulta los datos almacenados en el mecanismo de indexación a través de un servicio desarrollado en Symfony. Los datos son almacenados en el mecanismo de indexación luego del recorrido realizado por el mecanismo de rastreo. Esta arquitectura está basada en el patrón arquitectónico MVC, donde la función del modelo la tienen los servicios de Symfony los cuales son los encargados de implementar la lógica del negocio gestionando todos los accesos a la información almacenada en Solr.

2.7 Patrones de diseño

Los patrones de diseño representan la descripción de un problema particular y recurrente, que aparece en contextos específicos, y presenta un esquema genérico demostrado con éxito para su solución; este último se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí (Larman, 2004).

En el diseño de la herramienta de búsqueda de imágenes se tuvieron en cuenta los siguientes patrones GRASP (Patrones Generales de Software para Asignación de Responsabilidades), que describen los principios fundamentales de la asignación de responsabilidades a objetos:

Experto: este patrón plantea que se debe asignar una responsabilidad al experto en información, en otras palabras, a la clase que cuenta con los datos necesarios para cumplir la responsabilidad. De esta forma, se conserva el encapsulamiento de la información, puesto que los objetos ejecutan las tareas que le co-

rresponden de acuerdo a la información que poseen, lo que da lugar a sistemas más robustos y fáciles de mantener (Larman, 2004). En el marco de la presente investigación siguiendo el patrón Experto en Información se le asignaron responsabilidades determinadas solamente a las clases que cuentan con la información necesaria para dar cumplimiento a las mismas, esto se evidencia en la Tabla 8. De esta forma se mantiene el encapsulamiento de la información, puesto que los objetos utilizan su propia información para llevar a cabo las tareas. Normalmente, esto conlleva un bajo acoplamiento, lo que da lugar a sistemas más robustos y más fáciles de mantener.

Tabla 8: Relación de asignación de responsabilidades.

Clase objeto	**Responsabilidad**
ImageNakeDetection	Conocer si una imagen es apta o no para mostrar mediante la funcionalidad "Búsqueda segura" basado en el nivel de desnudo detectado en la imagen.
ImagePredominantColor	Conocer el color predominante en una imagen.
FaceRecognition	Conocer la cantidad de rostros que contiene una imagen.
SimpleMetadataExtractor	Conocer la cantidad de megapíxeles de la imagen, la categoría (pequeña, mediana o grande) y la proporción de la imagen respecto a sus dimensiones, y las dimensiones en cm de la imagen.
ImageParser	Conocer todas las características extraídas de la imagen para su envío al filtro de indización.

Creador: la instanciación de una clase es una de las actividades fundamentales en un sistema orientado a objetos. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, con lo que se logra menos dependencia y mayores oportunidades de reutilización de código (Larman, 2004). La clase "ImageParser" es la encargada de analizar el contenido de la imagen pasada en el parámetro "content", para esto crea instancias de las clases "ImagePredominantColor", "FaceRecognition" e ImageNakeDetection, estas realizan el procesamiento necesario para detectar el color predominante de la imagen, la cantidad de rostros que contiene y el nivel de desnudos que presenta respectivamente. La Figura 9 muestra el flujo de la creación de los objetos.

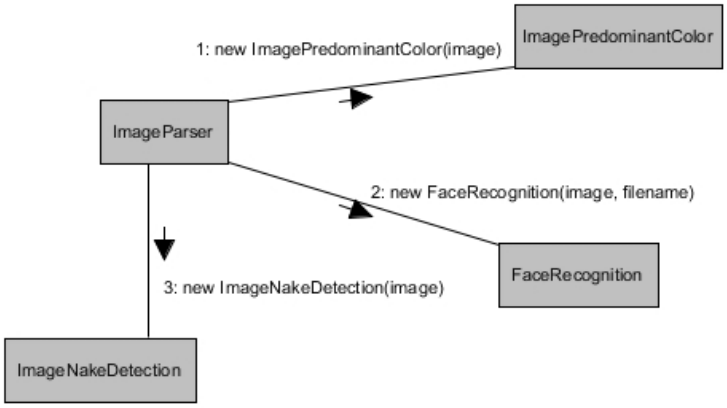


Figura 9: Creación de objetos en la clase “ImageParser”.

Alta cohesión: en el diseño orientado a objetos, la cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento (clase o subsistema). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que colaboran entre sí y con otros objetos para simplificar su trabajo. Una clase con alta cohesión es relativamente fácil de mantener, entender y reutilizar (Larman, 2004). La utilización de este patrón de diseño se manifiesta en el diseño de la clase “ImageParser”. Esta clase en orden de cumplir la responsabilidad que le es asignada mediante el patrón Experto, colabora y a su vez delega responsabilidades en las clases “SimpleMetadataExtractor”, “FaceRecognition”, “ImageNakeDetection” e “ImagePredominantColor”. La utilización de este patrón se evidencia además, en el diseño de las clases “ImageNakeDetection” e “ImagePredominantColor”, las cuales colaboran y delegan la responsabilidad de las operaciones en los espacios de color en la clase “ColorSpaceOperation”. En el Anexo 4 se muestra un fragmento del diagrama de secuencia del CU “Procesar imágenes”, donde se evidencia el uso de este patrón.

Controlador: este patrón tiene como objetivo asignar la responsabilidad a una clase de recibir o manejar un mensaje de evento del sistema generado por un actor externo, por lo general a través de una interfaz gráfica de usuario a la que accede un usuario para realizar ciertas operaciones en el sistema (Larman, 2004). La utilización de este patrón se evidencia en la clase *“InterfacesController”*, la misma se encarga de atender y ofrecer respuesta a cada una de las peticiones realizadas por el usuario mediante la interfaz web. El siguiente fragmento del Diagrama de Secuencia perteneciente al CU “Buscar imagen de forma avanzada”, muestra el flujo de peticiones recibidas por esta clase, así como las respuestas.

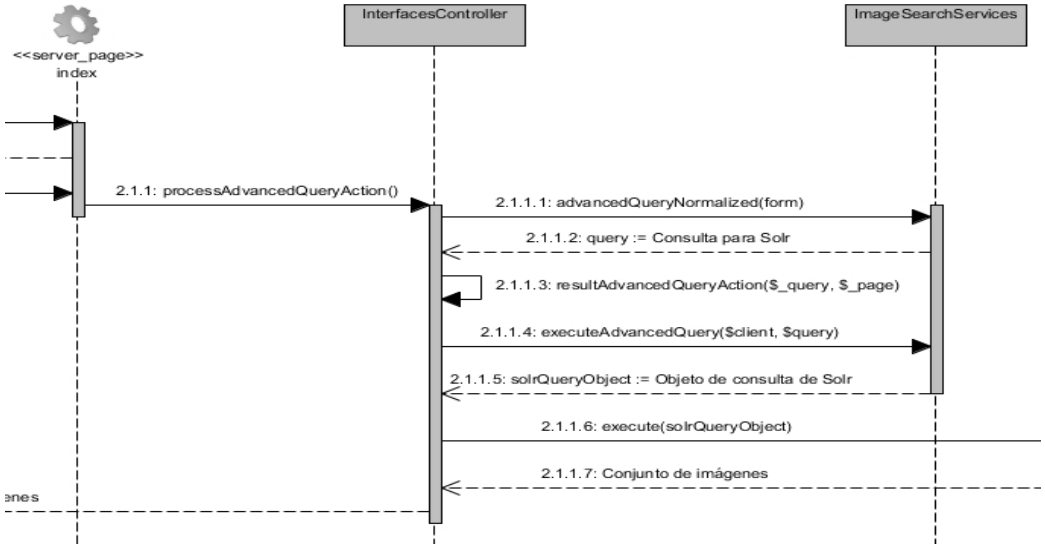
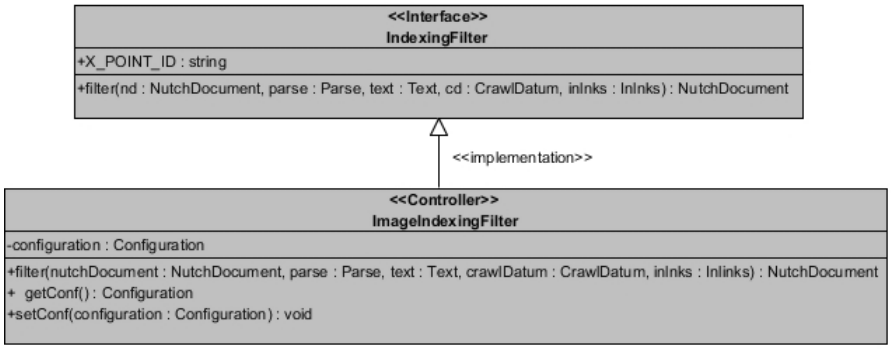


Figura 10: Fragmento del Diagrama de Secuencia perteneciente al CU1 “Buscar imagen de forma avanzada”.

2.8 Modelo de Diseño

El modelo de diseño es aquel que se encarga de describir la realización de los casos de uso del sistema, y se utiliza como medio de abstracción del modelo de implementación y el código fuente del software. Su objetivo fundamental es transmitir, a través de la representación mediante diagramas, una comprensión en profundidad de los aspectos relacionados con los requerimientos no funcionales y restricciones concernientes a los lenguajes de programación (Larman, 2003).

2.8.1 Diagrama de clases del diseño



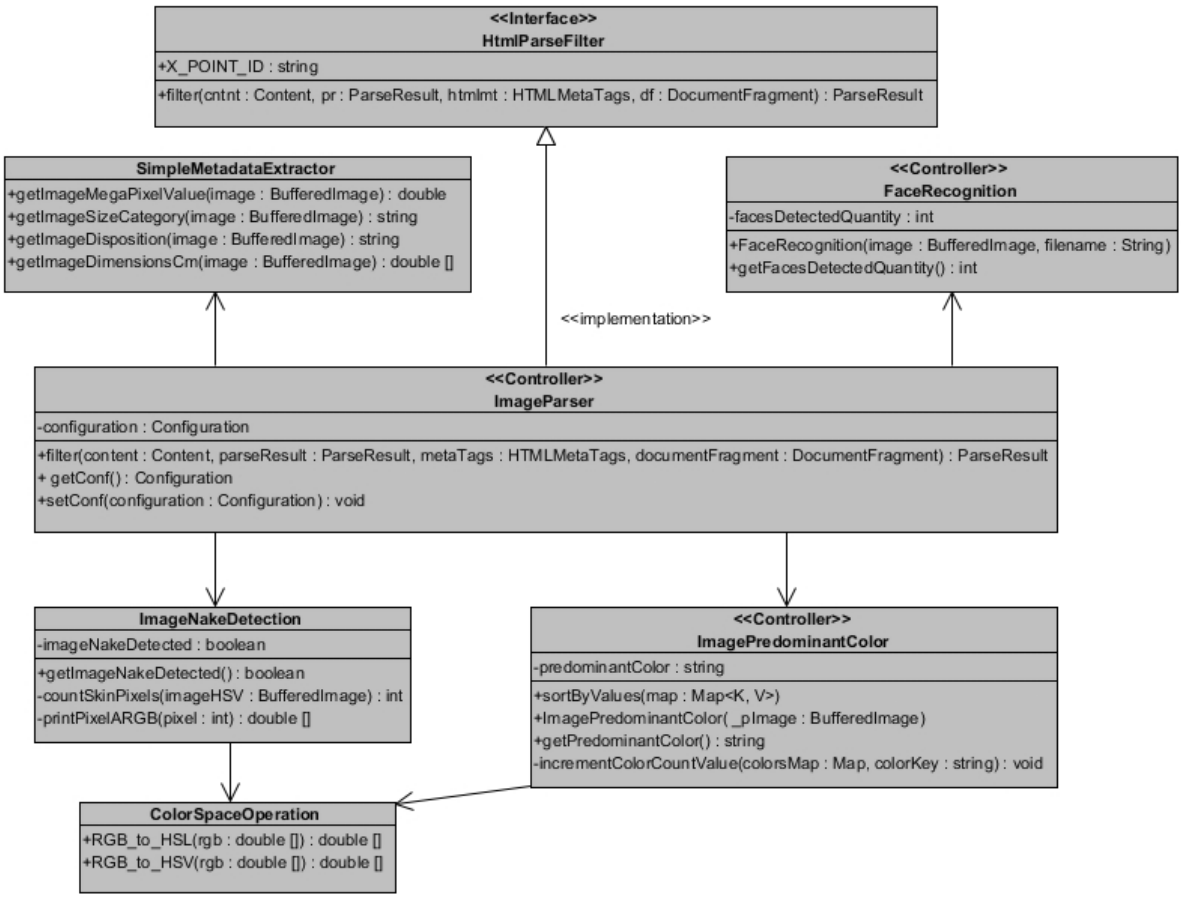


Figura 11: Diagrama de clases del diseño CU “Procesar imagen”.

En la Figura 11, se muestran las clases que intervienen en el procesamiento de una imagen. Para esto, el rastreador, una vez realizado el recorrido y recuperados los documentos, implementa una fase para analizar los documentos, donde la clase HtmlParserFilter procesa el contenido de cada documento haciendo uso de las bibliotecas incorporadas. Durante este proceso la clase “ImageParser” es la encargada de extraer los metadatos de la imagen, para ello esta clase depende de las clases “SimpleMetadataExtractor”, “FaceRecognition”, “ImageNakeDetection” e “ImagePredominantColor”, las cuales son las responsables de: obtener los metadatos básicos tales como: tamaño, cantidad de megapíxeles, disposición, dimensiones, entre otros; determinar si la imagen contiene rostros o no; determinar el nivel porcentual de desnudos presentes en la imagen; y determinar el color predominante en la misma respectivamente. Además, las dos últimas clases mencionadas dependen de la clase “ColorSpaceOperation” para la realización de

cálculos y conversiones entre el espacio de color de la imagen y el necesario para el procesamiento de la misma. Una vez terminado el procesamiento el documento es procesado por los filtros de indexación que implementan la interfaz “IndexingFilter”, en este interviene la clase “ImageIndexingFilter” para definir cuáles de los metadatos extraídos por “ImageParser” serán indexados, y cómo. El resto de los diagramas de clases del diseño se pueden ver en el Anexo 5.

2.9 Diagrama de interacción

Los diagramas de interacción son utilizados para modelar los comportamientos dinámicos que caracterizan un sistema informático. Estos suelen representar un conjunto de objetos o clases y sus relaciones, así como los mensajes que se pueden enviar entre ellos.

2.9.1 Diagramas de secuencia

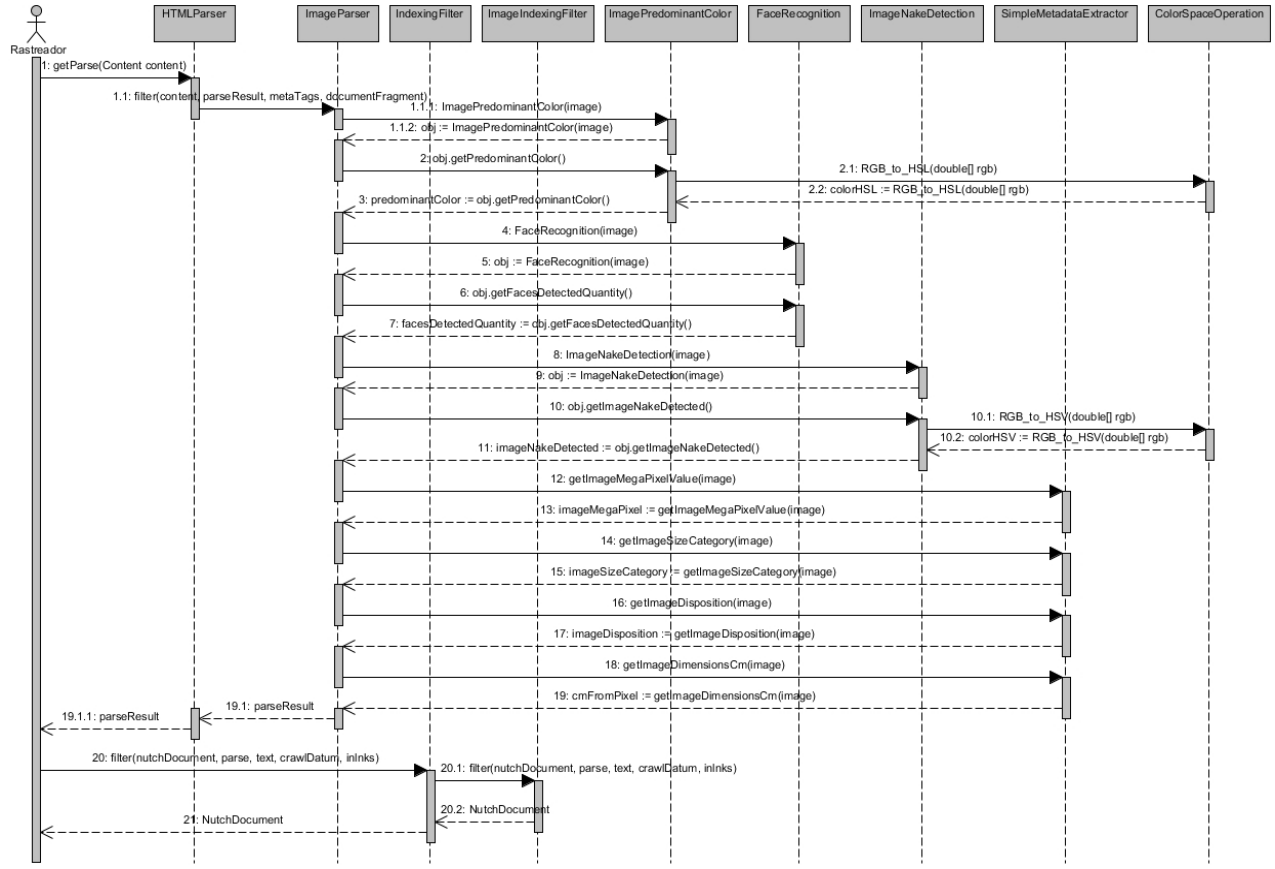


Figura 12: Diagrama de secuencia del CU “Procesar imagen”.

Con el propósito de modelar el comportamiento dinámico del sistema, se presenta en la Figura 12. el diagrama de secuencia por cada escenario del CU “Procesar imagen” descrito con anterioridad. El resto de los diagramas de secuencia se encuentran en el Anexo 6.

2.10 Modelo de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se relacionan en esos nodos (SparxSystems, 2014).

Como se puede apreciar en la Figura 13, el nodo “Dispositivo-Cliente” representa un dispositivo utilizado por el usuario desde el cual se podrán realizar las búsquedas de imágenes, a través del protocolo HTTP, haciendo uso de un navegador web. El nodo “Servidor de aplicaciones web Apache” es el encargado de atender y ofrecer respuesta a cada una de las solicitudes del cliente. Además, se observan dos nodos más, uno que representa el servidor maestro para Solr con Tomcat7 como contenedor de servlets y otro donde deberá ser instalado el servidor maestro Nutch. Se sugiere que se encuentren en servidores independientes con el objetivo de utilizar al máximo las características de hardware y software de estos, aunque pudieran alojarse en uno solo.

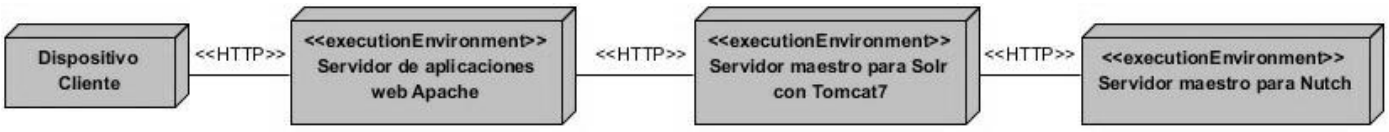


Figura 13: Diagrama de despliegue de la herramienta de búsqueda de imágenes.

2.11 Conclusiones parciales

En este capítulo se abordaron una serie de aspectos correspondientes al análisis y diseño de la herramienta de búsqueda de imágenes digitales publicadas en la Web llegándose a la siguiente conclusión:

- La representación y descripción de los artefactos generados garantizaron un mejor entendimiento de los flujos de trabajos presentes en el proceso de búsqueda de una imagen.
- La especificación de los requisitos funcionales y no funcionales del sistema, dieron paso a una mejor comprensión, por parte de los autores, de los resultados que se pretenden obtener de una manera precisa y sirvieron de guía para la implementación del sistema.

- La definición de la arquitectura y los patrones de diseño a utilizar, permitieron establecer las bases para fomentar la reutilización y las buenas prácticas de programación entre los desarrolladores durante la fase de implementación, así como disminuir el impacto de los cambios futuros en el código fuente.
- La elaboración del diagrama de despliegue permitió identificar la disposición física de los artefactos de la herramienta informática a desarrollarse.

CAPÍTULO: 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA HERRAMIENTA DE RECUPERACIÓN DE IMÁGENES DIGITALES PUBLICADAS EN LA WEB

3.1 Introducción

La implementación del sistema es una de las fases imprescindibles dentro del proceso de desarrollo de software. Esta fase comprende la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuestos en la etapa de análisis y diseño; con el objetivo de conformar el producto final requerido por el cliente (Larman, 2004).

Aparejado al proceso de implementación, el software que se construye debe ser sometido a determinadas pruebas que corroboren la correspondencia entre el producto y los requisitos definidos en las etapas anteriores. A esta etapa se le conoce como validación del sistema y en ella, pueden realizarse diferentes tipos de pruebas en función de los objetivos de las mismas.

3.2 Modelo de componentes que integran la herramienta informática

El modelo de componentes representa la forma en que es estructurado un sistema informático atendiendo a las diferentes partes que lo componen. Partiendo de este punto, (Sommerville, 2005) puntualiza que cada componente debe ser tratado como una unidad de composición independiente e indispensable dentro de un sistema, y que puede contraer relaciones de dependencia con otros componentes. Algunos ejemplos de componentes físicos lo constituyen los archivos, módulos, librerías, ejecutables, binarios, entre otros.

3.2.1 Diagrama de componentes

Un diagrama de componentes permite visualizar con facilidad la estructura general del sistema y el comportamiento de las funcionalidades que estos componentes proporcionan y utilizan a través de la interfaces. Además, muestra la organización y las dependencias entre un conjunto de componentes. Ver Figura 14 y Figura 15.

A continuación se describen los principales paquetes que componen el diagrama de componentes correspondiente a la interfaz web de la herramienta implementada:

- **InterfaceBundle:** Agrupa en su interior todos los componentes de la interfaz web, estableciendo una estructura organizativa acorde al patrón de arquitectura MVC.
- **Controller:** Contiene la clase controladora encargada de procesar las peticiones de las páginas clientes y del usuario según Figura 8: Arquitectura del sistema. y devolver las respuestas con la información requerida.
- **Form:** Alberga las clases para construir los formularios utilizados para el envío de información por parte de los usuarios.
- **Config:** Contiene en su mayoría los archivos donde se definen las rutas de la aplicación.
- **Public:** En su interior contiene diferentes componentes de apoyo a las vistas de la aplicación entre otros recursos. Dentro de este paquete se encuentran los archivos CSS (estilos de la herramienta), los archivos JavaScripts (archivos para aplicar dinamismo a partes del código HTML) y las imágenes.
- **Views:** Agrupa las páginas referentes a las vistas de la aplicación, así como las plantillas bases.

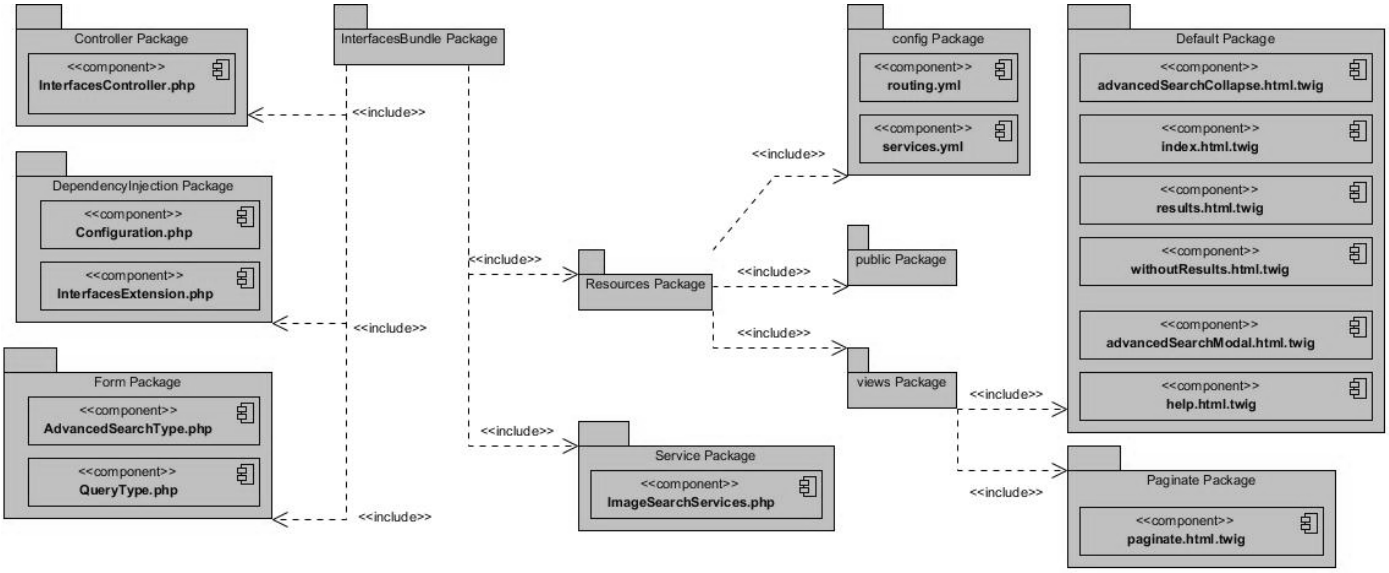


Figura 14: Diagrama de componentes correspondiente a la interfaz web de la herramienta de recuperación de imágenes digitales publicadas en la Web.

A continuación se describen los principales paquetes que componen el diagrama de componentes correspondiente al plugin de Nutch, que recupera y procesa las imágenes digitales publicadas en la Web:

- **ImageLibrary:** Agrupa en su interior todos los componentes asociados al plugin implementado, estableciendo una estructura organizativa como la que tienen los proyectos java.
- **org.apache.nutch.parse.image:** Posee las clases principales encargadas de la extracción de los metadatos de la imagen.
- **org.apache.nutch.parse.image.auxiliar:** Componente que agrupa las clases que brindan funcionalidades a las clases encargadas de la extracción de los metadatos de la imagen.

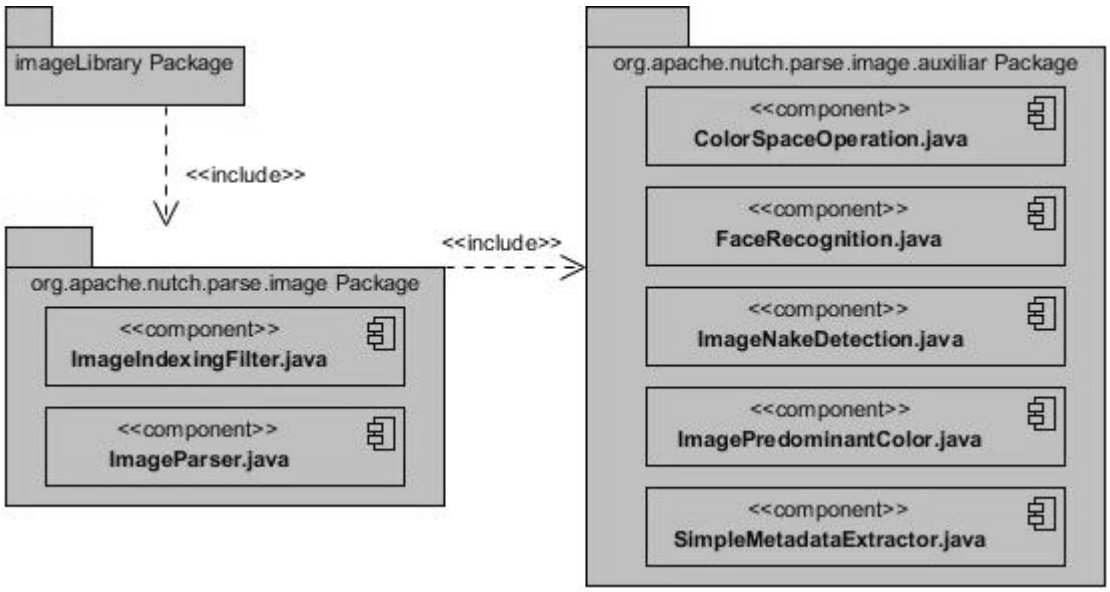


Figura 15: Diagrama de componentes correspondiente al plugin de Nutch que recupera y procesa las imágenes en la herramienta de recuperación de imágenes digitales publicadas en la Web.

3.3 Estándares de codificación utilizados

Los estándares de codificación son especificaciones o estilos que establecen la forma de generar el código funcional de las aplicaciones informáticas. Puesto que en muchas ocasiones, los sistemas de cómputo son implementados por varios programadores, la adopción inicial de un único estilo de codificación constituye uno de los factores de mayor peso en la calidad, rendimiento, legibilidad y capacidad de mantenimiento del producto final.

El estándar de codificación utilizado en el lenguaje PHP es el que establece el marco de trabajo Symfony2 que sigue los estándares definidos en los documentos PSR-1, PSR-2, PSR-3, PSR-4 (SensioLabsNetwork, 2014; PHP Framework Interop Group (psr-1), 2014; PHP Framework Interop Group (psr-2), 2014; PHP Framework Interop Group (psr-4), 2014). Por otra parte el estándar de codificación utilizado para el

lenguaje Java es el que establece la comunidad de Nutch para su uso. Entre sus elementos más relevantes y comunes se encuentran:

- Añadir un espacio después de cada delimitador coma ´, ´.
- Añadir un único espacio a ambos lados de un operador: =, ==, && etc.
- En los array multilínea, añade una coma al final de cada elemento, incluido el último.
- Añade un salto de línea antes de una sentencia return, a menos que el return se encuentre solo en un bloque de sentencias, y un salto después de cada llave cierre de sentencia, excepto después de la llave de cierre de clase.
- Usa notación camelCase sin guiones bajos en variables, funciones, métodos y argumentos.
- El nombre de las clases se realiza en UpperCamelCase, es decir, que comienza por mayúscula.

3.4 Validación de la herramienta de recuperación de imágenes digitales publicadas en la Web

A continuación se detallan los tipos de pruebas de software aplicadas a la herramienta implementada. Las mismas persiguen como objetivo fundamental, la detección de las no conformidades respecto a las funcionalidades de la aplicación, la medición del grado de usabilidad de las funcionalidades implementadas, así como también la correcta integración entre los diferentes componentes de la arquitectura del sistema.

3.4.1 Pruebas funcionales

Las pruebas funcionales son aquellas que se aplican a un software determinado, con el objetivo de validar que las funcionalidades implementadas funcionen de acuerdo a las especificaciones de los requisitos definidos con anterioridad. Para la ejecución de este tipo de pruebas, suelen emplearse dos métodos fundamentales: el método de Caja Blanca y el método de Caja Negra. El primero se centra en las pruebas al código de las aplicaciones; mientras que el segundo permite a los probadores enfocar su atención en el funcionamiento de la interfaz, a través del análisis de los datos de entrada y los de salida. En este epígrafe se exponen los aspectos concernientes a las pruebas funcionales realizadas utilizando el método de Caja Negra, a partir de los casos de prueba diseñados.

Diseño de los casos de prueba basados en casos de uso

Se realizaron un total de 6 casos de prueba basados en casos de uso cubriendo la totalidad de los casos de uso de la investigación. A continuación se muestra un fragmento del diseño del caso de prueba basado en caso de uso “Procesar imagen”. El resto de los diseños de casos de prueba se pueden encontrar en el

Anexo 8.

Tabla 9: Caso de prueba correspondiente al CU "Procesar imagen"

Escenario	**Descripción**	**V1**	**V2**	**V3**	**Respuesta del sistema**	**Flujo central**
EC 1.1 Procesar imagen.	Se le envía al sistema la URL de la imagen para ser procesada.	https://dragones.uci.cu/media/k2/items/cache/5762a31292907de83a148ef235752b83_L.jpg	NA	NA	Metadatos de la imagen.	1- Crear una carpeta con el nombre “urls” dentro de la raíz del rastreador. 2.- Crear un fichero de texto con un nombre aleatorio y escribir dentro de él la URL de la imagen que se describe en la variable 1. 4.- Ejecutar el comando: “bin/crawl-imagenes urls/ crawl/ ruta_Solr 2”, donde ruta_Solr es la ruta del servidor de Solr.
EC 1.2 Filtrar una imagen para ser indexada.	Se le envía al sistema toda la información obtenida de una imagen para que la filtre y posteriormente pueda ser indexada por el rastreador.			Contiene los siguientes metadatos: **megaPixel:** 72800 **transparency:** false **predominantColor:** blue **fileSize:** 25 **fullTitle:** 5762a31292907de83a148ef235752b83_L **imageSizeCategory:** pequeña **imageProportion:** ancha **widthCm:** 9,3 **heightCm:** 5,6 **width:** 350 **height:** 208	Objeto que contiene todos los metadatos descritos en las variables 3, los cuales son los que se van a indexar.	1.- Crear una carpeta con el nombre “urls” dentro de la raíz del rastreador. 2.- Crear un fichero de texto con un nombre aleatorio y escribir dentro de él la URL: http://internos.uci.cu/cine/reposici-n 4.- Ejecutar el comando: “bin/crawl-imagenes urls/ crawl/ ruta_Solr 2”, donde ruta_Solr es la ruta del servidor de Solr.

Tabla 10: Variables empleadas en el diseño del caso de prueba basado en el CU “Procesar imagen”.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	content	Variable JAVA (instancia de la clase org.apache.nutch.protocol.Content)	No	Contiene información referente al contenido de la página que se está analizando; tal como: URL, el binario del contenido, metadatos propios del protocolo de la URL y el tipo de contenido.
2	nutchDocument	Variable JAVA (instancia de la clase org.apache.nutch.indexer.NutchDocument)	No	Contiene los metadatos del documento que se va a indexar.
3	parse	Variable JAVA (instancia de la clase org.apache.nutch.parse.Parse)	No	Contiene los metadatos extraídos durante el análisis de la imagen.

Resultados de las pruebas funcionales

Con el objetivo de probar el correcto funcionamiento de las funcionalidades del sistema se realizaron tres iteraciones de pruebas a la herramienta. En la Tabla 11 se muestran los resultados obtenidos en cada iteración de prueba a la herramienta de recuperación de imágenes digitales publicadas en la Web, así como la corrección de cada uno de los errores.

Tabla 11: Cantidad de no conformidades por cada iteración de las pruebas funcionales.

No conformidades	1ra Iteración	2da Iteración	3ra Iteración
Detectadas	6	4	2
Resueltas	6	4	2
Pendientes	0	0	0

En la Figura 16 se puede apreciar el comportamiento de las no conformidades de las pruebas funcionales ejecutadas.

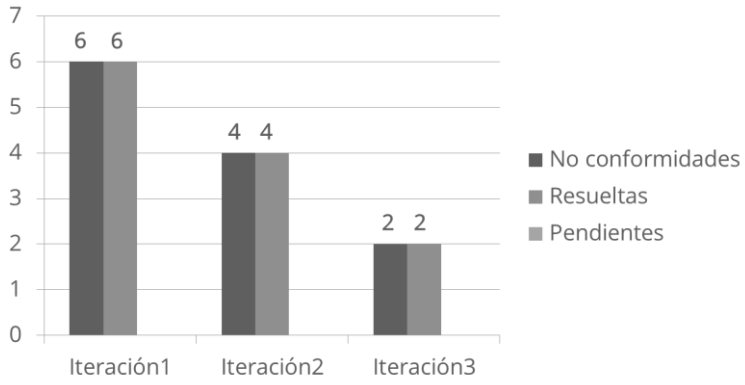


Figura 16: Comportamiento de las no conformidades por cada iteración de las pruebas funcionales.

3.4.2 Pruebas de integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Las mismas validan que estos componentes realmente funcionan juntos, son llamados correctamente y además, transfieren los datos correctos en el tiempo preciso y por las vías de comunicación establecidas (Sommerville, 2005).

Una vez realizadas las pruebas funcionales a cada componente interno de manera independiente, y verificado que las funcionalidades implementadas se corresponden de acuerdo a los requisitos funcionales y no funcionales establecidos; se pudo comprobar el correcto funcionamiento de los componentes mediante el estudio del flujo de datos entre ellos. Posterior a estas pruebas, se hace necesaria la realización de pruebas de integración, con la finalidad de validar la compatibilidad y el funcionamiento de las interfaces que comunican las diferentes partes que componen la solución informática.

Para la realización de las pruebas de integración se llevaron a cabo diferentes acciones, a continuación se mencionan las fundamentales:

- Verificación de la conexión de Nutch y Solr para indexar los metadatos extraídos de la imagen. Ver Anexo 8, Caso de prueba: “Integración Nutch-Solr”.
- Comprobar el enlace entre Solr y la interfaz de resultados para verificar si se muestran todas las imágenes almacenadas en Solr. Ver Anexo 8, Caso de prueba: “Integración Solr-Interfaz web”.

La ejecución de las pruebas de integración permitió verificar el trabajo conjunto de los componentes de la herramienta en cuestión. Se hizo énfasis en la integración entre la interfaz web, el mecanismo de indexación Solr y el mecanismo de rastreo Nutch, para detectar incoherencias en el funcionamiento de la aplicación, no encontrándose ninguna no conformidad, llegándose a la conclusión que existe una correcta integración entre los componentes internos de la herramienta.

3.4.3 Pruebas de carga y estrés

Las pruebas de carga consisten en probar el funcionamiento del software bajo condiciones extremas. Estudia la especificación del software, las funciones que debe realizar, las entradas y las salidas, analizando los valores límites (López, 2010). Las pruebas de estrés están diseñadas para enfrentar al programa a condiciones anormales. Las pruebas ejecutan un sistema, de manera que demande recursos en cantidad, frecuencia o volúmenes extremos.

Para la herramienta de recuperación de imágenes digitales publicadas en la Web es preciso realizar dichas pruebas pues resulta necesario comprobar el rendimiento del sistema soportando una cantidad máxima de usuarios que soliciten este recurso en la Web. Para la realización de las mismas se utiliza la herramienta JMeter en su versión 2.12.

La prueba realizada consistió en definir 2 pruebas de 200 y 300 hilos de concurrencia cada una, las cuales simulan 200 y 300 accesos de usuarios respectivamente. La primera prueba fue realizada en un servidor Pentium(R) Dual-Core a 2.20Ghz de velocidad de procesador, una memoria RAM de 4.00Gb y a las 21:07 hs para 200 hilos, cada “1 seg”. La segunda prueba fue realizada en una PC Core-i3 a 2.4Ghz de velocidad de procesador, una memoria RAM de 6.00Gb, a las 12:30hs configurando 300 hilos, cada “1 seg”.

Para un mejor entendimiento de las componentes “Reporte resumen” que se verán a continuación, se explica cada parámetro que la compone.

- **#Muestras:** cantidad de hilos utilizados para la URL.
- **Media:** tiempo promedio en milisegundos para un conjunto de resultados.
- **Min**: tiempo mínimo que demora un hilo en acceder a una página.
- **Max:** tiempo máximo que demora un hilo en acceder a una página.
- **Rendimiento:** rendimiento medido en los requerimientos por segundo / minuto / hora.
- **Kb/sec:** rendimiento medido en Kbytes por segundo.

Los valores totales obtenidos por la componente “Reporte resumen” para 200 hilos se muestran en la Tabla 12.

Tabla 12: Resultados de prueba de carga y estrés con el acceso de 200 usuarios.

	#Muestras	Media	Min	Max	%Error	Rendimiento	Kb/sec
TOTAL	5425	2792	1	65170	0	31.683/segundos	2602.274

Como se muestra en la Tabla 12, la herramienta desarrollada, para 200 usuarios conectados de forma concurrente respondió 5 425 peticiones al servidor en un promedio de 2.792 segundos, lo que equivale a 31.683 peticiones por segundo.

Los valores totales obtenidos por la componente “Reporte resumen” para 300 hilos se muestran en la Tabla 13.

Tabla 13: Resultados de prueba de carga y estrés con el acceso de 300 usuarios.

TOTAL	#Muestras	Media	Min	Max	%Error	Rendimiento	Kb/sec
	10200	4605	2	132957	0.029	47.2/segundos	6659.84

Como se muestra en la Tabla 13, la herramienta desarrollada, para un total de 300 usuarios conectados de forma concurrente respondió 10 200 peticiones al servidor en un promedio de 4.605 segundos, lo que equivale a 47.2 peticiones por segundo.

De manera general estos resultados son favorables teniendo en cuenta que cumplen el RNF 16, siendo este un resultado satisfactorio para los autores de la presente investigación.

3.4.4 Pruebas de seguridad

La seguridad informática comprende la puesta en práctica de un conjunto de medidas preventivas y reactivas en los sistemas informáticos y tecnológicos, que posibilitan la protección de la información, persiguiendo como objetivo principal la integridad, confidencialidad y disponibilidad de la misma (INTECO-CERT, 2014).

La realización de pruebas de seguridad contribuye a la detección temprana de vulnerabilidades y la toma de medidas para la disminución de amenazas de ataque, y con ello proveer sistemas de cómputo más seguros y confiables. A la herramienta desarrollada se le realizaron una serie de pruebas de seguridad mediante el software Acunetix, las cuales se presentan a continuación:

- Ataques de inyección.
- Cross-Site Scripting (XSS).
- Falsificación de petición (CSRF).
- Detección de ficheros y directorios.

Se encontraron 16 alertas en total, de ellas ninguna de clasificación alta, 3 de clasificación media, 4 de clasificación baja y 9 informacionales. De manera general los resultados obtenidos se presentan en la Tabla 14 y 15.

Vulnerabilidades encontradas

Tabla 14: Vulnerabilidades del entorno

Tipo	Cantidad	Descripción	Recomendaciones
Software	1	La versión de PHP instalada contiene vulnerabilidades que han sido solucionadas en versiones más recientes.	Actualizar a la versión 5.6.4 o superior.
Configuración	13	La configuración del servidor permite la captación de información sensible por el atacante.	Esta vulnerabilidad aunque no permite al atacante tomar el control acerca de la información a obtener puede proporcionar al atacante información sensible que puede utilizar para realizar ataques más específicos, pueden solucionarse limitando el acceso a los archivos del servidor y los elementos de la configuración ajustando el "expose_php" en “off” en el archivo php.ini.

Tabla 15: Vulnerabilidades del sistema

Tipo	Cantidad	Descripción	Recomendaciones
Falso positivo	1	El sistema permite la inyección de código.	Esta vulnerabilidad no puede ser utilizada por el atacante para alterar el sistema ni obtener información por lo que se denomina falso positivo.

Observaciones: Todas las vulnerabilidades fueron resueltas en el servidor y se recomienda que se tenga en cuenta la nueva configuración en el servidor para un futuro despliegue del sistema.

Los resultados obtenidos en las pruebas de seguridad, avalados por el centro de calidad UCI, fueron satisfactorios, llegando a la conclusión que la herramienta desarrollada es segura y está en condiciones de ser usada por el cliente con el propósito de satisfacer las necesidades de búsqueda de imágenes en la Web.

3.4.5 Validación de la hipótesis de la investigación

Con el propósito de evaluar los indicadores referentes a **porción del documento indexado, exhaustividad, precisión, tasa de fallo e índice de irrelevancia** en la herramienta de recuperación de imágenes digitales, correspondiente a la variable dependiente definida como parte de la hipótesis de investigación, se realizó un experimento donde se compararon los resultados obtenidos en el buscador Orión actual y en la herramienta implementada. Para el experimento se crearon 3 colecciones con 100

imágenes cada una, donde de las 100, 30 respondían al criterio de búsqueda y las otras 70 no. Para evaluar el indicador **porción del documento indexado** se compararon los metadatos indexados en cada una de las herramientas, y para los 4 indicadores restantes se realizaron 3 iteraciones de pruebas con diferentes criterios de búsqueda. En las tablas siguientes se representan los resultados generales obtenidos por cada indicador medido.

Tabla 16: Resultados de la medición del indicador “Porción del documento indexado”.

Buscadores/Indicador	**Metadatos indexados**
Herramienta desarrollada	29
Orión	11

Tabla 17: Resultados de la medición del indicador “Exhaustividad”.

Buscadores/Criterios	**Criterio 1**	**Criterio 2**	**Criterio 3**
Herramienta desarrollada	93%	86.7%	96.7%
Orión	10%	33.3%	N/A

Tabla 18: Resultados de la medición del indicador “Precisión”.

Buscadores/Criterios	**Criterio 1**	**Criterio 2**	**Criterio 3**
Herramienta desarrollada	87.5%	78.8%	96.7%
Orión	75%	76.9%	N/A

Tabla 19: Resultados de la medición del indicador “Tasa de fallo”.

Buscadores/Criterios	**Criterio 1**	**Criterio 2**	**Criterio 3**
Herramienta desarrollada	5.7%	10%	1.4%
Orión	1.4%	4.3%	N/A

Tabla 20: Resultados de la medición del indicador “Índice de irrelevancia”.

Buscadores/Criterios	**Criterio 1**	**Criterio 2**	**Criterio 3**
Herramienta desarrollada	4%	7%	1%
Orión	1%	3%	N/A

Al interpretar las tablas se debe tener en cuenta que los resultados arrojados por Orión a partir de un criterio de búsqueda eran pocos, lo que provocó que el resultado del indicador **precisión** en algunos casos fuera tan parecido al de la herramienta desarrollada y que además, **tasa de fallo** e **índice de irrelevancia** fueran tan bajos. (Ver fórmulas para el cálculo de indicadores en el Anexo 9). A partir del análisis de la comparación entre las tablas y teniendo en cuenta lo anteriormente explicado, se evidencia que utilizando la herramienta desarrollada se obtienen resultados con mayor calidad de relevancia para el usuario con

respecto a Orión, siendo esto un resultado satisfactorio que apoya la hipótesis de la presente investigación.

3.5 Conclusiones parciales

En este capítulo se abordaron una serie de aspectos correspondientes a la implementación y validación de la herramienta de búsqueda de imágenes digitales publicadas en la Web llegándose a la siguiente conclusión:

- La representación y descripción del diagrama de componentes permitió visualizar con más facilidad la estructura general de la herramienta.
- La ejecución de pruebas a la herramienta permitió detectar las deficiencias presentes, subsanarlas en el menor tiempo posible y ofrecer una aplicación con mayor calidad, seguridad y usabilidad.
- La aplicación del método experimental y la realización de cálculos estadísticos, aportaron elementos sustanciales que permitieron validar la hipótesis de investigación planteada con anterioridad, y con ello la factibilidad de la herramienta implementada.

CONCLUSIONES

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

- A partir del estudio realizado de los fundamentos teóricos relacionados con la recuperación de información y procesamiento de imágenes se determinó que existen una serie de Sistemas de Recuperación de Información los cuales brindan funcionalidades que implican un procesamiento previo de las imágenes, no encontrándose ninguna de licencia libre para su uso, definiéndose una propuesta de solución de acuerdo a las necesidades existentes.
- El enfoque ágil propuesto por la metodología OpenUp y el uso de las tecnologías y herramientas seleccionadas, permitieron analizar y describir los subprocesos que se debían ejecutar, concretando así, en concordancia con las especificaciones del cliente, las características que debía tener la herramienta a desarrollarse.
- Una vez estudiados los elementos que intervienen en el proceso de recuperación de imágenes digitales en la Web, fue posible la modelación de los artefactos que contribuyeron al diseño de la propuesta de solución posibilitando un mayor soporte a la implementación de los requisitos previamente expresados por el cliente; garantizando la estructura base para la organización lógica del código fuente y la disminución del impacto ante futuras modificaciones en la aplicación.
- La implementación de la herramienta informática para la recuperación de imágenes digitales utilizando las herramientas y tecnologías estudiadas y orientada por las cuatro fases de la metodología OpenUp permitió solucionar los problemas existentes planteados en la problemática de la presente investigación.
- La evaluación de las pruebas de software realizadas permitió erradicar las insuficiencias detectadas en la herramienta desarrollada logrando así un producto más seguro y funcional conforme a las necesidades de los usuarios finales.
- La validación de la hipótesis a través de un experimento demostró la calidad de la herramienta desarrollada, resultado que fue satisfactorio para los autores quedando demostrada la hipótesis de la investigación.

RECOMENDACIONES

Una vez concluida la investigación y el desarrollo de la propuesta de solución, los autores del presente trabajo recomiendan:

- Implementar funcionalidades para la búsqueda de imágenes a partir de una imagen introducida como criterio de búsqueda.
- Añadir funcionalidades a la herramienta que permitan la búsqueda semántica.
- Integrar la herramienta de recuperación de imágenes digitales en el entorno real de Orión y Red Cuba.

REFERENCIAS BIBLIOGRÁFICAS

AKSYNOFF, A. *Sphinx 2.3.2-dev reference manual*. [En línea]. Sphinx | Open Source Search Server, 2014. [Citado el: 14 de Noviembre de 2014.]. Disponible en: [http://sphinxsearch.com/docs/current.html].

ACUNETIX. *Audit your website security with Acunetix Web Vulnerability Scanner.* [En línea]. Acunetix, 2015. [Citado el: 26 de Mayo de 2015.]. Disponible en: [http://www.acunetix.com/].

APACHE SOFTWARE FOUNDATION. *The Apache Solr Reference Guide.* [En línea]. Apache Solr – Resources, 2014. [Citado el: 22 de Octubre de 2014.].
Disponible en: [http://lucene.apache.org/solr/documentation.html].

ACOSTA, J.; GREINER, C.; DAPOZO, G.; ESTAYNO, M. *Medición de atributos POO en frameworks de desarrollo PHP.***.** Buenos Aires, Argentina : s.n., 2014.

BAENA, J. D.; TREVILLA, I.; RIVAS, S. *Librerías de Procesamiento y Síntesis de Imágenes*. [PDF]. Repositorio Institucional, Universidad de Sevilla, 2014. [Citado el: 14 de Noviembre de 2014.] Disponible en:
[http://munkres.us.es:8080/sandbox/groups/catam/wiki/a8b29/attachments/78dba/SAV.Pres.LibreriasProcesamientoImagenes.VFinal.pdf?sessionID=5fdf778a1ecc958bc3eac704b19ccb519d1668fa].

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern information retrieval.* New York: ACM Press; Harlow [etc.]: Addison-Wesley, 1999. ISBN 0-201-39829-X.

BAHIT, E. *POO y MVC en PHP. El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC.* [En línea]. Slideshare, 2011. [Citado el: 20 de Enero de 2015.] Disponible en: [http://www.slideshare.net/eugeniabahit/poo-y-mvc-en-php-por-eugenia-bahit].

BAKKEN, S. S. *Manual de PHP.* s.l. : The PHP Documentation Group, 2013. pág. 1063.

BARKOV, A. *mnoGoSearch 3.3.15 reference manual: Full-featured search engine software* .[En línea]. 2014. [Citado el: 13 de Octubre de 2014.] Disponible en: [http://www.mnogosearch.org/doc33/msearch-intro.html#features].

COMECHE, J., A., M. *Los modelos clásicos de recuperación de información y su vigencia.* [PDF].e-printsin library & informtaion science, 2014.[Citado el: 2 de Octubre de 2014]. Disponible en: [http://eprints.rclis.org/9662/].

CASTELLS, M. *Internet y la sociedad en red.* [En línea]. Lliçó inaugural del programa de doctorat sobre la societat de la informació i el coneixement, 2013. [Citado el: 29 de Mayo de 2015]. Disponible en: [http://www.uoc.edu/web/cat/articles/castells/castellsmain12.html].

CUBANIC. 2014. *Estadísticas sobre: ¿Cuántos dominios hay bajo .cu?*.[En línea].CUBANIC, Portal Cuba .cu, 2014. [Citado el: 7 de Febrero de 2015.]. Disponible en: [http://www.nic.cu/estadisticas.php].

CARRILLO, G.; RAMÍREZ, Y. *Colocándonos en la web.* [En línea].Instituto Internacional de Periodismo

"José Martí". Diplomados, Cursos, Talleres para periodistas de Cuba y A. Latina, 2012. [Citado el: 10 de Mayo de 2014.]. Disponible en: [https://periodismojosemarti.wordpress.com/2012/11/28/colocandonos-en-la-web/].

CAMARGO, F., I.; SALINAS, S., O. *Evolución y tendencias actuales de los Web crawlers.* 2, 2013, Vol. 18.

DANS, E. Information Management. [En línea]. 2006. [Citado el: 15 de Septiembre de 2014.]. Disponible en: [http://informationmanagement.wordpress.com/category/gestion/gestion].

DEPARTMENT OF COMPUTER SCIENCES UNIVERSITY OF CHILE. *WIRE - Web Information Retrieval Environment.* [En línea]. WIRE (Web Information Retrieval Environment): Center for Web Research, 2011. [Citado el: 11 de Diciembre de 2014.]. Disponible en: [http://www.cwr.cl/projects/WIRE/]

DB-ENGINES. *DB-Engines Ranking - Trend of Elasticsearch Popularity.* [En línea]. Historical trend of Elasticsearch popularity, 2014. [Citado el: 14 de Enero de 2015.] Disponible en: [http://db-engines.com/en/ranking_trend/system/Elasticsearch].

DB-ENGINES. *System Properties Comparison Elasticsearch vs. Solr vs. Sphinx.* [En línea]. Elasticsearch vs. Solr vs. Sphinx Comparison, 2015. [Citado el: 22 de Marzo de 2015.] Disponible en: [http://db-engines.com/en/system/Elasticsearch%3BSolr%3BSphinx].

DB-ENGINES-RANKING. *DB-Engines Ranking.* [En línea]. Historical trend of the popularity ranking of database management systems, 2014. Disponible en: [http://db-engines.com/en/ranking].

DÍAZ, R., G. *Estudio de la incidencia del conocimiento lingüistico en los Sistemas de Recuperación de Información.* Universidad de Salamanca. Salamanca : s.n., 2002.

EGUILUZ, J. *Desarrollo web ágil con Symfony2,* 2013. pág. 618.

ELASTICSEARCH. *Apache Solr vs ElasticSearch*. [En línea]. Apache Solr vs ElasticSearch - the Feature Smackdown!, 2014. Disponible en: [http://solr-vs-elasticsearch.com/].

ELLIS, D. *Progress and problems in information retrieval. London.* Los Angeles : s.n., 1996.

DE LA FRAGA, L., G. *Cinvestav.* [PDF]. Conferencias impartidas, 2001. [Citado el: 16 de Septiembre de 2014.] Disponible en: [http://delta.cs.cinvestav.mx/~fraga/Charlas/procImagen.pdf].

FRAMEWORK PROCESS ECLIPSE. eclipse.org. [En línea] 2014. [Citado el: 24 de Octubre de 2014.] . Disponible en: [http://epf.eclipse.org/wikis/openupsp/openup_basic/customcategories/introduction_to_openup_basic,_BTJ_YMXwEduywMSzPT].

GOOGLE INC. *Annual Report*. [En línea], Google Inc. Form 10-K Annual Report Filed, 2013. [Citado el: 24 de Octubre de 2014.]. Disponible en: [http://edgar.secdatabase.com/1404/119312513028362/filing-main.htm].

GORMLEY, C.; TONG, Z.. *Elasticsearch: The Definitive Guide.* [En línea]. Elastic, 2014. [Citado el: 11 de Noviembre de 2014.] Disponible en: [http://www.elasticsearch.org/guide/].

GILL, T.; GILLILAND., A., J.; WHALEN M.; WOODLEY M., S. *Introduction to Metadata,* Getty Publications, 2008. págs 96.

HERNÁNDEZ, M., C. *Manual de Usuario del Motor de Búsqueda Cubano.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2013.

HERRERA, A., G., L. *Modelos de Sistemas de Recuperación de Información Lingüística Difusa.* 2006.

JALÓN, J., G.; RODRÍGUEZ, J., I.; MINGO, I.; IMAZ, A.; BRAZALÉZ, A.; LARZABAL, A.; CALLEJA, J. *Aprenda Java como si estuviera en primero.* Navarra : s.n., 2000.

KORFHAGE, R. R. *Information Storage and Retrieval. New York.* 1997.

KUMAR, S P. *Integration of Web mining and web crawler: Relevance and State of Art.* s.l. : International Journal on Computer Science and Engineering, 2010. págs. 772-776. Vol. 2. ISSN: 0975-3397.

LARMAN, C. *UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado.* Segunda. s.l. : Prentice Hall, 2004. pág. 520.

LIE, H., W.; BOS, B. *Cascading Style Sheets – designing for the Web*, ISBN: 0321193121, 2005.

LAPUENTE, M., J. *Metadatos para imágenes.* [En línea]. 2014. [Citado el: 22 de Marzo de 2015.] Disponible en: [http://www.hipertexto.info/documentos/metad_imag.htm].

LÓPEZ, E., S. *Ecología y libertad*. [En línea]. ATIX, Revista Digital, 2010. [Citado el: 14 de Mayo de 2015.]. Disponible en: [http://osl.ugr.es/descargas/atix16.pdf].

MARIÑO, C., V. *Programación en PHP5.Nivel básico.* 2008.

MÉNDEZ, F., J., M. *Recuperación de información: modelos, sistemas y evaluación.* Murcia : EL KIOSKO JMC, 2004. 84-932537-7-4.

MÉNDEZ, F., J., M. *Propuesta y desarrollo de un modelo para la evaluación de la recuperación de información en internet.* Tesis Doctoral, Universidad de Murcia, Facultad de Ciencias de la Documentación, España, 2002.

MEXIDOR, D. F. *Ciberguerra contra Cuba: Mentiras en la red.* [En línea]. Cubadebate contra el terrorismo mediático, 2011. [Citado el: 28 de Noviembre de 2014.] Disponible en: [http://www.cubadebate.cu/opinion/2011/03/22/ciberguerra-contra-cuba-mentiras-en-la-red/].

MARTÍNEZ, W., B. *Impactos del bloqueo Yanqui en la UCI.* [En línea]. Cubadebate contra el terrorismo mediático, 2013. [Citado el: 6 de Mayo de 2015.] Disponible en: [https://cubaxdentro.wordpress.com/2013/10/25/impactos-del-bloqueo-yanqui-en-la-uci/]**.**

MEDINA, R.; BELLERA., J. *BASES DEL PROCESAMIENTO DE IMÁGENES MÉDICAS.*[PDF]. 2014. [Citado el: 25 de Octubre 2014]. Disponible en: [http://www.saber.ula.ve/redtelemedicina/TallerTelemedicina/j_bellera-01.pdf].

NETCRAFT. *Web server survey*. Sitio web de NetCraft. [En línea]. 2015. Disponible en: [http://news.netcraft.com/].

NETMARKETSHARE. Market Share Statistics for Internet Technologies. [En línea]. Market share for mobile, browsers, operating systems and search engines | NetMarketShare, 2014. [Citado el: 8 de Noviembre de 2014.] . Disponible en: [www.netmarketshare.com].

NIETO, I., A., M. *Universidad Nacional de Colombia.* [En línea] 2009. [Citado el: 10 de Octubre de 2014.] . Disponible en: [http://dis.unal.edu.co/profesores/eleon/cursos/tamd/presentaciones/nutch.pdf].

NUGRAHA, A. *Indexing Bibliographic Database Content Using MariaDB and Sphinx Search Server.* [En línea] . The Code4Lib Journal – Indexing Bibliographic Database Content Using MariaDB and Sphinx Search Server, 2014. [Citado el: 25 de Octubre de 2014.] Disponible en: [http://journal.code4lib.org/articles/9793].

ORACLE. Sitio oficial del IDE NetBeans. [En línea]. NetBeans IDE Features, 2014. [Citado el: 24 de Octubre de 2014.]. Disponible en: [https://netbeans.org/features/index.htm].

OTTO, M.; THORNTON, J. *Bootstrap3 Manual Oficial.* [trad.] Javier Eguiluz. 2013

POTENCIER, F. *What is Symfony2*?. [En línea]. SensioLabsNetwork, 2011. [Citado el: 18 de Abril de 2015.] Disponible en: [http://fabien.potencier.org/article/49/what-is-symfony2].

PUGH, D., S., E. *Apache Solr 3 Enterprise Search Server.* s.l. : Pack Publishing, 2009. pág. 418. ISBN 978-1-84951-606-8.

PHP FRAMEWORK INTEROP GROUP (PSR-1). *Codificación estándar básica.* [En línea]. www.php-fig.org, 2014. [Citado el: 26 de Mayo de 2015.] Disponible en: [http://www.php-fig.org/psr/psr-1/es/].

PHP FRAMEWORK INTEROP GROUP (PSR-2). *Codificación estándar básica.* [En línea]. www.php-fig.org, 2014. [Citado el: 26 de Mayo de 2015.] Disponible en: [http://www.php-fig.org/psr/psr-2/es/].

PHP FRAMEWORK INTEROP GROUP (PSR-4). *Codificación estándar básica.* [En línea]. www.php-fig.org, 2014. [Citado el: 26 de Mayo de 2015.] Disponible en: [http://www.php-fig.org/psr/psr-4/].

REYNOSO, C.; KICILLOF, N. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* Buenos Aires : s.n., 2004.

ROSSINI, D. *Los archivos y la nuevas tecnologías de la información*.[PDF] 2003. [Citado el: 1 de Noviembre de 2014.]. Disponible en: [http://eprints.rclis.org/4651/1/lapaz11.pdf].

RUEDAS, E., R.; DELGADO, Y., H. *Los spider en función de los motores de búsqueda*.[PDF] .uciencia,

2012, pág. 12.

REAL ACADEMIA ESPAÑOLA. *Diccionario de la lengua española*. [En línea]. Real Academia Española, 2014. [Citado el: 15 de Septiembre de 2014.] . Disponible en: [http://lema.rae.es/drae/?val=informaci%C3%B3n].

SÁNCHEZ, B., S. *La importancia de lo visual (Un ejemplo con fotografías).* [En línea] 2014. [Citado el: 22 de Octubre de 2014.]. Disponible en: [http://cvc.cervantes.es/ensenanza/biblioteca_ele/asele/pdf/08/08_0757.pdf].

SENSIOLABSNETWORK. *Coding Standards.* [En línea] SensioLabs Product, 2014. [Citado el: 26 de Mayo de 2015]. Disponible en:[http://symfony.com/doc/current/contributing/code/standards.html].

SALOMÓN, O., P. *Los planes de Google y el acceso a Internet en Cuba*. [En línea]. Cubadebate contra el terrorismo mediático, 2014. [Citado el: 2 de Mayo de 2015.]. Disponible en: [http://www.cubadebate.cu/opinion/2014/07/04/los-planes-de-google-y-el-acceso-a-internet-en-cuba/]

SÁNCHEZ, J. *Manual de referencia JavaScript*. [PDF]. 2003. [Citado el: 6 de Octubre de 2014.]. Disponible en: [http://www.jorgesanchez.net/web/javascript.pdf].

SPARKS, G. *Introducción al modelado de sistemas de software usando el Lenguaje Unificado de Modelado (UML). .* Australia : s.n., 2013.

SETA, L., D. *Apache Solr: una introducción*. [En línea]. Apache Solr: una introducción - Dos Ideas. 2010. [Citado el: 3 de 10 de 2014.]. Disponible en: [http://www.dosideas.com/noticias/java/913-apache-solr-una-introduccion.html].

SOLR-WIKI. *Public Websites using Solr*. [En línea]. PublicServers - Solr Wiki, 2014. [Citado el: 4 de Noviembre de 2014.] Disponible en: [https://wiki.apache.org/solr/PublicServers].

SOMMERVILLE, I. *INEGINERÍA DE SOFTWARE.* Madrid : Pearson Educación S.A, 2005. 84-7829-074-5.

SPARXSYSTEMS. *Diagrama de Despliegue UML 2.* [En línea]. Sparx Systems - Tutorial UML 2 - Diagrama de Despliegue, 2014. [Citado el: 24 de Febrero de 2015.] Disponible en: [http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html].

SPHINX. *System Properties Comparison Elasticsearch vs. Solr vs. Sphinx.* [En línea] 2014. [Citado el: 14 de Febrero de 2015.] Disponible en: [http://db-engines.com/en/system/Elasticsearch%3BSolr%3BSphinx].

SLIDESHARE. *Comparing open source search engines.* [En línea]. Slideshare, 2014. [Citado el: 11 de Marzo de 2015.]. Disponible en:[http://www.slideshare.net/rboulton/comparing-open-source-search-engines].

THE APACHE SOFTWARE FOUNDATION-TIKA. Apache Tika. [En línea]. Apache Solr, 2014. [Citado el: 13 de Octubre de 2014.] Disponible en: [http://tika.apache.org/].

THE APACHE SOFTWARE FOUNDATION-SOLR. Apache Solr. [En línea]. Apache Solr , 2014. [Citado el: 3 de Octubre de 2014.] Disponible en: [http://lucene.apache.org/solr/].

THE APACHE SOFTWARE FOUNDATION-JMETER. Apache JMeter. [En línea]. Apache JMeter, 2015. [Citado el: 24 de Marzo de 2015.] Disponible en: [http://jmeter.apache.org/].

VISUAL PARADIGM. Visual Paradigm for UML - Software design tools for agile software development. [En línea] 2014. [Citado el: 24 de Octubre de 2014.] . Disponible en: [http://www.visual-paradigm.com/product/vpuml/].

WAINSCHERNKER, R., S.; MASSA, J.; TRISTAN, P. *ETAPAS DEL PROCESAMIENTO DIGITAL DE IMÁGENES.* 2011, Guía informativa área Procesamiento de Señales.

WORLD WIDE WEB CONSORTIUM. *HTML/Specifications***.** [En línea] 2014. [Citado el: 24 de Octubre de 2014.] .Disponible en: [http://www.w3.org/community/webed/wiki/HTML/Specifications#HTM].