Universidad de las Ciencias Informáticas

Facultad # 1

Título: *Buscador de la Red Social de la Universidad de las Ciencias Informática*

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

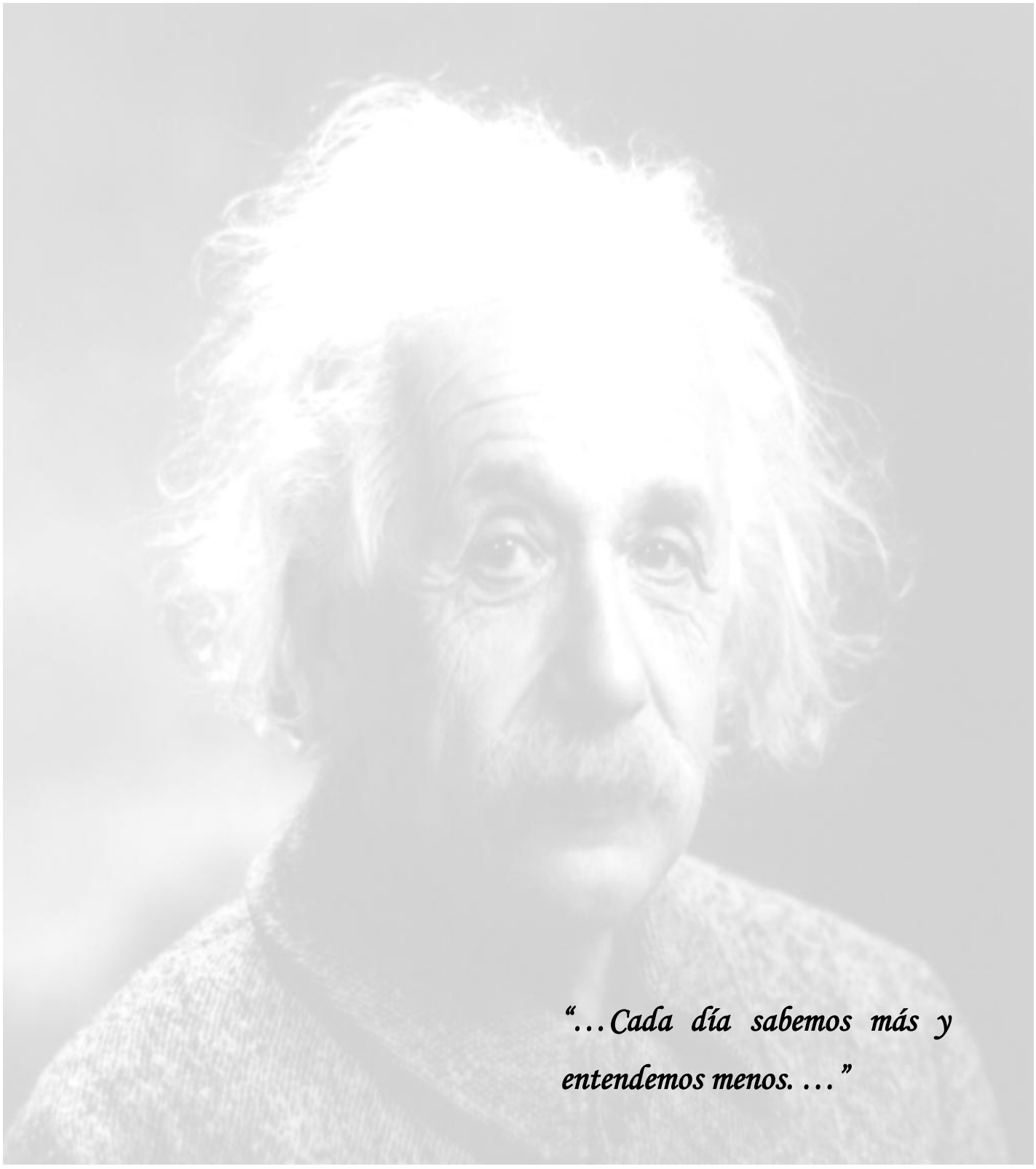
Autores: *Yoennis Palmero Marrero*

Tutores: *Ing. Yusliel García Vázquez*

Ing. Damarys Cano López

La Habana, Junio de 2012.

"Año 54 de la Revolución."



"…Cada día sabemos más y entendemos menos. …"

Declaro ser autor de la presente tesis y reconozco al Centro de Informatización Universitaria y a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yoennis Palmero Marrero

_____

Firma del Autor

Ing. Yusliel García Vázquez

_____

Firma del Tutor

Ing. Damarys Cano López

_____

Firma del Tutor

Ing. Yusliel García Vázquez: Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI) desde el 19 de Julio del 2007. Al graduarse pasa a ser profesor de la Universidad de las Ciencias Informáticas, en las disciplinas de Práctica Profesional y Programación. Ha impartido las asignaturas de Práctica Profesional 1, Programación 2, Principios de Algoritmización, Introducción a la Programación, Programación 1 y Programación 3. Ha realizado trabajos de desarrollo de *software* para los eventos del Fórum de Ciencia y Técnica y en cual también se ha desempañado como Presidente de las Comisiones a nivel de Base. Ha participado en eventos como UCIENCIA. Ha ejercido como tribunal de tesis de pregrado. Ha trabajado en proyectos productivos relacionados con la creación de aplicaciones *web* para el país y para el extranjero. Está matriculado en la maestría de Informática Aplicada, de la cual ya cursó el Diplomado Básico y el Diplomado Especializado. Se desempeñó como Líder de Equipo para la implementación del Portal Institucional del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela y del Sistema de Procesamiento de Opiniones que se desarrolló para el Centro de Estudios Sociopolíticos y de Opinión del CC PCC. Actualmente es desarrollado de la línea de Soluciones para Redes Sociales y Comercio Electrónico del Centro de Informatización Universitaria. **Correo electrónico:** ygarciav@uci.cu.

Ing. Damarys Cano López: graduado de Ingeniero en Ciencias Informáticas con Título de Oro en la Universidad de las Ciencias Informáticas en junio del 2011. Pertenece al departamento de Universidad Digital del Centro de Informatización Universitaria de la Facultad 1. **Correo electrónico:** dcano@uci.cu.

...A mis padres Carmen y Rosell, a quienes les debo la vida y viviré eternamente agradecido por inculcarme siempre sus sabios consejos y experiencias, por confiar en mi incluso primero que yo, por su amor y sacrificio para que yo pudieran hacer realidad este sueño de hacerme ingeniero...

...A mi hermana Roxana por darme la oportunidad de servirle de guía y ejemplo...

...A mis abuelos por toda la confianza que han depositado en mí...

...A mis tíos y tías por siempre estar presente cuando los necesitaba...

...A todos mis amigos que vienen desde primer año soportándome…

… A mi novia Rosa por compartir estos 19 meses y 7 días de felicidad, de comprensión y de amor incondicional I love you…

...A mis amigos Yannier, Jose, Alien, Yoanis y Cheyla por estar presente en las buenas y en las malas...

.. A mis tutores Yuslier y Damarys por regañarme y aconsejarme, por estar siempre encima de mí todo el tiempo sin dejarme relajar, por servirme de guía de forma incondicional…

… A Novoa, Paul y los muchachones de CIDI que me ayudaron en la realización de la tesis, día a día sin importar la hora o lo ocupados que estaban, muchas gracias...

... A todos aquellos profesores que de una forma u otra me enseñaron y me ayudaron a llegar a ser un ingeniero...

...A las personas que de una forma u otras aportaron su granito de arena para que lograra hacer este sueño realidad y nunca dejaron de confiar en mí...

...A la UCI por permitir hacer realidad mi mayor sueño que es el de hacerme ingeniero...

Muchas gracias.

… A mi mamá y papá, porque son los culpables de hacer de mí un mejor hombre, porque son el mejor regalo que me ha dado la vida, mi mayor alegría, mis ojos, mi gran virtud, son la mayor muestra de amor, sacrificio y bondad que Dios ha guardado para mí. Ustedes son lo que más amo…

… A mi hermana Roxana por brindarme en todo momento su cariño y amor…

… A mis abuelos que han hecho de mí un hombre de bien, que han vivido para mí y darían con los ojos cerrados una vida de felicidad por evitarme un instante de sufrimiento. Siempre los llevo en mi corazón...

… A mi novia Rosa por amarme con mis defectos y virtudes, me ha enseñado a tener paciencia, a ser fuerte y a saber que todo es posible. Te amo…

… A Dios por darme la fe, sabiduría y salud, por permitir que mis seres queridos me vean realizado mi sueño de convertirme en ingeniero…

RESUMEN

El presente trabajo tiene como objetivo desarrollar un buscador en la Red Social de la Universidad de las Ciencias Informáticas que permita la recuperación de la información existente en la misma. Para dar cumplimiento al objetivo planteado en la investigación, se realiza un estudio del estado del arte referente a los principales motores de búsqueda e indexadores existentes a nivel nacional e internacional y se describen las funcionalidades y servicios que brindan. Se investigan el proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI, las tecnologías y herramientas utilizadas para el desarrollo de la propuesta de solución. Para lograr un mayor acercamiento al tema se exponen los elementos fundamentales de la arquitectura, patrones de diseño y estándar de código. Se realiza la implementación de la solución propuesta, obteniéndose un producto de *software* para lograr o dar cumplimiento al objetivo planteado inicialmente. Para la evaluación del buscador de la Red Social de la Universidad de las Ciencias Informáticas, se emplea una metodología de evaluación de sistema de recuperación de información, además se realizan pruebas para verificar el correcto funcionamiento de la aplicación de acuerdo a los requisitos planteados.

Palabras clave: búsqueda, contenido *web*, información, recuperación.

Índice de figuras

Índice de tablas

Introducción

Desde su aparición, la *World Wide Web* (WWW o simplemente *Web*), se ha convertido en un instrumento de uso cotidiano por la sociedad. La generación dinámica de páginas, la conexión con bases de datos, la interactividad con el usuario, la usabilidad, entre otras, son algunas de las tendencias evolutivas que han marcado el desarrollo de la *Web* en los últimos años. La *Web* ha cambiado profundamente la forma de comunicación, de hacer negocios y de trabajar, se pueden realizar transacciones económicas a través de la *Web*, tener acceso a millones de recursos, independientemente de la situación geográfica e idioma.

En la actualidad, el crecimiento de la información presente en Internet, que abarca una buena parte del saber humano, se comporta de manera exponencial. En muchos casos, esta información se encuentra dispersa y poco estructurada lo que hace muy engorroso el proceso de encontrar información útil en la red.

Las redes sociales son formas de interacción social compuestas por grupos de personas, las cuales están conectadas por uno o varios tipos de relaciones, tales como amistad, parentesco, intereses comunes o que comparten conocimientos, siendo la *web*, la herramienta que permite dichas interacciones, llevando las relaciones personales al plano de las comunicaciones digitales o virtuales.

La Universidad de las Ciencias Informáticas (UCI), proyecto creado en el 2002 por el Comandante en Jefe Fidel Castro Ruz, no se encuentra ajena al uso de las redes sociales. "*Mientras que el empleo de este tipo de tecnología por parte de los usuarios UCI es elevado, la presencia de la universidad como institución en las mismas es casi nula. La institución cuenta con un grupo en Facebook*[1] *que no es mantenido por ningún departamento de la institución, sino que fue creado por los propios estudiantes y trabajadores en función de contar con un espacio de debate social con el cual sentirse identificados".*(Clavijo 2011)

Las redes sociales, no son solo los sitios con intenciones transparentes e ingenuas que permiten comunicarse con los amigos de modo desinteresado, sino que además constituyen una herramienta inteligente de obtención de información de los usuarios de Internet que representa una fortaleza para las instituciones dueñas de la información y una debilidad para el eslabón más débil de la misma: los usuarios. Sin embargo las redes sociales, con el enfoque apropiado, pueden ser una herramienta muy útil, para fomentar las investigaciones y la colaboración científica, aumentar el trabajo comunitario con una tendencia creciente al uso y desarrollo de servicios o convertirse en un complemento para el proceso docente-educativo. Por lo antes planteado, en la UCI se propuso desarrollar una red social de

[1] **Facebook:** sitio oficial, http://www.facebook.com.

carácter académico que ofrece a sus usuarios las mismas funcionalidades que brindan las redes sociales convencionales existentes en la actualidad y otras más de corte académicos en beneficio de la institución.

Actualmente, la *Web* es un espacio diseñado para el intercambio de información. La cual ha generado la necesidad de compartir información entre distintas instituciones o estructuras de una misma organización, sin embargo, la heterogeneidad y descentralización de las fuentes de información que la *Web* presenta han provocado que mientras más información hay accesible, más difícil es localizar lo que se busca. Los Sistemas de Recuperación de Información (SRI), constituyen el mecanismo ideal para resolver este tipo de problemas, estos permiten localizar y procesar la información de forma rápida y automática. Son sistemas capaces de localizar cualquier contenido existente en la *Web*, tales como textos, imágenes, videos, archivos de sonido, entre otros. Por lo anterior expuesto, se ha popularizado el uso de los motores de búsqueda como Google[2], Yahoo[3] y Altavista[4], que permiten a través de algunas palabras o términos de búsqueda retornar un conjunto de páginas *web* como resultado.

El desarrollo económico, tecnológico y social de una nación, está cada vez más ligado al desarrollo del conocimiento científico y tecnológico de la misma. En Cuba, se lleva a cabo un profundo proceso de informatización de la sociedad. En el marco de este proceso, se ha implementado un motor de búsqueda llamado *2x3*[5], cuyo objetivo es dotar a la red nacional de una herramienta que permita realizar búsquedas en todos los sitios cubanos. Este proyecto ha sido desarrollado por la Oficina Nacional para la Informatización (ONI), entidad con significativos aportes a la informatización del país, que en estos momentos esta oficina se encuentra trabajando en conjunto con la UCI, específicamente con el Centro de Ideo-informática (CIDI).

En la UCI la gran cantidad de sitios *web* existentes provoca que muchas veces los usuarios se desorienten a la hora de buscar información de algún tema de interés. La falta de información ya no es un problema, lo que muchas veces ocurre es que los usuarios no saben dónde se encuentra la información deseada.

Dentro de la universidad son varios los trabajos que se han desarrollado en vista de lograr la eficiente recuperación de información dispersa en los sitios publicados dentro de la entidad, con el despliegue

[2] **Google:** sitio oficial, http://www.google.com.

[3] **Yahoo:** sitio oficial, http://www.yahoo.es.

[4] **Altavista:** sitio oficial, http://www.altavista.com.

[5] **2x3:** sitio oficial, http://www.2x3.cu.

de los mismo no se han obtenidos resultados positivos, en este caso se encuentran los motores de búsqueda Faro y Orion, los cuales se encuentran en estos momentos, descontinuados, desactualizados o sin soporte. En la actualidad las personas dentro de la organización no pueden realizar una búsqueda sobre toda la información publicada en la red y los usuarios se ven en la necesidad de consumir su cuota de navegación por hacer uso de los buscadores internacionales (Google, Yahoo), para encontrar información sobre algún tema de interés, cuando en ocasiones resulta que la misma se encuentra publicada en la red de la propia entidad.

Esta situación conduce al siguiente **problema a resolver** ¿Cómo recuperar y mostrar la información sobre los recursos dispersos y publicados en la Red Social de la Universidad de las Ciencias Informáticas?

Para darle solución a este problema, el **objeto de estudio** de la investigación lo constituyen los sistemas de búsqueda y recuperación de la información y como **campo de acción** está encaminado hacia el buscador de la Red Social de la Universidad de las Ciencias Informáticas.

Dadas las condiciones mencionadas con anterioridad se plantea como **objetivo general:** desarrollar una herramienta que permita recuperar la información existente en la Red Social de la Universidad de las Ciencias Informáticas mediante la utilización de un motor de búsqueda y auxiliándose de las páginas *web* publicadas en la universidad.

Del objetivo general planteado se derivan los siguientes **objetivos específicos:**

- Caracterizar aspectos teóricos conceptuales sobre los sistemas de búsqueda y recuperación de la información en una red informática.
- Desarrollar un buscador basado en palabras claves para la Red Social de la Universidad de las Ciencias Informáticas.
- Validar la solución propuesta a partir de la realización de pruebas al sistema.

Para desarrollar la investigación se emplearon los siguientes **métodos científicos:**

Métodos teóricos:

Analítico / Sintético: se evidencia en el profundo estudio de los sistemas de búsqueda de información existentes a nivel nacional e internacional, a partir de la cual, para facilitar su entendimiento, se dividieron cada uno de los aspectos investigados, guiando la búsqueda hacia las temáticas más importantes, permitiendo así la extracción de los elementos más significativos y el arribo a conclusiones teóricas y prácticas bien definidas.

Histórico / Lógico: se evidencia en el análisis detallado del comportamiento de los buscadores, así como sus características y funcionalidades.

Métodos empíricos:

Observación: se hace un estudio minucioso de trabajos realizados con anterioridad que han logrado resultados excelentes y que se encuentran estrechamente vinculados con la problemática a la que se hace referencia.

Entrevista: mediante la entrevista con personas prácticas del tema al que se hace referencia en la presente investigación se podrán obtener ideas lógicas y precisas sobre la base de lo que se desea proponer.

Al concluir la investigación se **espera obtener** un buscador basado en palabras claves, partiendo de la información disponible en los sitios *web* existentes en la UCI y su integración con la Red Social de la Universidad de las Ciencias Informáticas.

El presente trabajo consta de una introducción, tres capítulos, conclusiones generales, referencias bibliográficas, bibliografía consultada y glosario de términos. A continuación se describen los capítulos que forman al siguiente trabajo:

Capítulo I. Fundamentación teórica del buscador de la Red Social de la Universidad de las Ciencias Informáticas: en este capítulo se presentan los distintos sistemas de recuperación de información existentes en el ámbito internacional, nacional y dentro de los nacionales los que se han desarrollado en la UCI. Se realiza además un análisis de las herramientas y tecnologías a utilizar para el desarrollo del buscador para la Red Social de la Universidad de las Ciencias Informáticas, así como el proceso de desarrollo de *software* empleado para guiar el desarrollo del buscador.

Capítulo II. Propuesta de solución del buscador de la Red Social de la Universidad de las Ciencias Informáticas: en este capítulo se definen los requisitos de *software*, las estructuras de datos y los artefactos necesarios para la implementación de la solución propuesta. Además se exponen las principales características del sistema, su diseño, arquitectura y el estándar de código utilizado.

Capítulo III. Validación del buscador de la Red Social de la Universidad de las Ciencias Informáticas: se describen las pruebas realizadas a la propuesta de solución con el objetivo de verificar que cumple con todas las funcionalidades requeridas. Para esto se realizan pruebas guiadas por la metodología para la evaluación de los SRI, planteada en un trabajo de diploma realizado en la UCI.

Capítulo I. Fundamentación teórica del buscador de la Red Social de la Universidad de las Ciencias Informáticas

1.1 Introducción

En el presente capítulo se exponen varios criterios sobre la recuperación de la información. Se describen además, algunos de los motores de búsqueda existentes a nivel internacional y nacional, sus características. Se describen el proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI, tecnologías y herramientas empleadas durante el desarrollo de la investigación, además de la metodología empleada para evaluar la propuesta de solución.

1.2 Conceptos básicos de la *Web*

"A principios de la década de 1990, Tim Berners-Lee planteó la idea de crear un sistema global de información con el objetivo de fusionar las tecnologías de computadoras personales, para satisfacer la demanda de intercambio automático de información entre los científicos que trabajaban en diferentes universidades e institutos de todo el mundo". (CERN 2008) Con este sistema cada usuario en un ordenador podía navegar de forma automática por los nodos de una red, sin importar cómo funcionaban los sistemas de esos nodos. Este sistema global de información fue conocido como *World Wide Web*. En (GUTIÉRREZ 2008) se expresan las siguientes palabras de Berners-Lee: *"El concepto de la Web integró muchos sistemas de información diferentes, por medio de la formación de un espacio imaginario abstracto en el cual las diferencias entre ellos no existían. La Web tenía que incluir toda la información de cualquier tipo en cualquier sistema."*

La arquitectura lógica de la *Web* se sustenta en tres pilares fundamentales, los cuales son:

- **URI:** *Uniform Resource Identifier* (Identificador Uniforme de Recurso). Para poder referenciar a todos los documentos en la *Web* es necesario que cada uno tenga su nombre propio, el cual se conoce como URI. Una versión más elemental de URI es **URL**: *Uniform Resource Locator* (Localizador Uniforme de Recurso), el cual representa una dirección en la *Web*.
- **HTML:** *Hyper Text Markup Language* (Lenguaje de Marcas de Hipertexto). Lenguaje de marcado creado por Berners-Lee para la construcción de páginas *web*, a través del cual se define la estructura y el contenido de los documentos. Además de su simplicidad de uso, el hecho de ser un lenguaje de hipertexto permite re-direccionar al lector desde un punto cualquiera del texto a otro objeto en la *Web*. Estos puntos de re-direccionamiento son conocidos como *links* o enlaces.

- **HTTP:** *Hyper Text Transfer Protocol* (Protocolo para la Transferencia de Hipertexto). Permite enviar y recibir información desde un lugar a otro en la *Web*, logrando así la comunicación entre clientes y servidores *web*. El cliente puede ser un navegador *web* o un agente, mientras que el servidor es quien almacena o crea los recursos requeridos por el cliente, tales como documentos *HTML* o imágenes, por solo citar algunos.

"La popularidad de la Web como uno de los servicios de Internet ha provocado que en muchos casos sean confundidos ambos términos. Desde el punto de vista técnico son completamente diferentes, pues Internet es la red física que hace posible establecer la conexión entre distintos nodos, mientras que la Web representa la información que está publicada sobre dicha red. El crecimiento de la Web es exponencial, debido principalmente a la facilidad y libertad de publicar contenido; cualquier persona sin distinción de tipo o condición, que posea conocimientos básicos sobre el lenguaje HTML, puede crear y publicar una página web."(PÉREZ 2008)

Documento *web*

"Un documento web es un fichero identificado por una URL y puede representar, entre otros formatos, una imagen, un archivo multimedia, un archivo de texto, un documento en formato PDF[6], un archivo comprimido o una página web. Por su parte, una página web es un documento codificado en lenguaje HTML que puede contener enlaces a otros documentos por medio de sus respectivas URLs "(GUTIÉRREZ 2008).

Baeza-Yates en (GUTIÉRREZ 2008) define a la *Web* como compleja, basándose en las características de las páginas que la componen (ver **Figura 1**), expresando que existen páginas estáticas, dinámicas, públicas y privadas.

Las páginas estáticas son aquellas que están en un servidor *web* antes de ser solicitadas por un usuario, o sea, existen en todo momento; mientras que las páginas dinámicas no siempre están presentes en el servidor, sino que se crean justo en el instante que el usuario las solicita, principalmente por medio de una consulta al servidor o como resultado de llenar los datos de un formulario. Actualmente, la mayor parte de la *Web* está constituida por páginas dinámicas. Las páginas públicas son accesibles para todos los usuarios, en tanto las privadas son aquellas que están protegidas por contraseñas o están dentro de una Intranet.

[6] **PDF:** del inglés, *Portable Document Format,* (Formato de Documento Portátil).

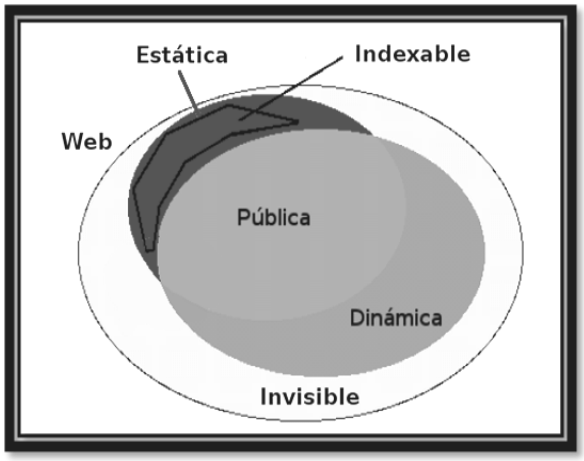


Figura 1: Principales características de la *Web*. Tomado de (Gutiérrez et al., 2008).

En (BOJO 2004) los autores definen también los conceptos de *Web* visible e invisible, definiendo la parte visible "*como aquellos documentos cuyo contenido puede ser mostrado a los usuarios por medio de buscadores web, mientras que la porción invisible se corresponde con los documentos no accesibles para estos buscadores, representando principalmente a contenidos recogidos en bases de datos u otros recursos informativos, los cuales necesitan ser consultados previamente para acceder a ellos, algo que los buscadores tradicionales no pueden realizar".*

1.3 Recuperación de la información

La recuperación de información es una actividad práctica que el ser humano realiza, consciente e inconscientemente, casi continuamente y en el marco de cualquier actividad. La necesidad de resolver una duda o documentar una afirmación, son expresiones clásicas de los procesos de recuperación de información. Con el desarrollo de los sistemas digitales de procesamiento de datos y tratamiento de la información, las técnicas de recuperación de información han ido desarrollando un conjunto de teorías y aplicaciones prácticas que subyacen en la actualidad a cualquier búsqueda y recuperación de información que tiene lugar en Internet.

Baeza-Yates en su artículo (Baeza-Yates 1999), considera la recuperación de la información como *"una necesidad de información (consulta) y un conjunto de documentos, ordenar los documentos más a menos relevantes para esa necesidad y presentar un conjunto de aquellos con mayor relevancia"*. Además presenta los principales problemas que, desde el punto de vista del contenido, enfrenta la recuperación de información:

- **Distribuido:** dada la estructura de la *Web*, los datos están en muchas computadoras y sobre distintas plataformas, la topología de la red no está predefinida y el ancho de banda y confiabilidad de las conexiones es muy variable.

- **Volátil:** los nombres de dominio y páginas aparecen y desaparecen de la red diariamente, además el volumen de los datos crece exponencialmente, doblando su tamaño en aproximadamente seis meses.
- **Dinámico:** actualmente la mayoría de las páginas se generan mediante una consulta a una base de datos y por ende es difícil recuperarlas sin conocer su estructura.
- **Redundante:** una porción de la *Web* está duplicada. La cantidad de información replicada, según Baeza-Yates en (GUTIÉRREZ 2008), "*es alrededor del 20% del total del contenido de la Web".*
- **Tipos heterogéneos:** existen variedad de medios digitales, de cada medio hay distintos formatos (por ejemplo, *HTML* o *Word* para texto, o *JPG* [7] y *GIF*[8] para imágenes), existen además diferentes lenguajes y alfabetos.
- **Calidad heterogénea:** la *Web* es un medio de publicación en el que los contenidos no pasan ningún tipo de proceso editorial, por lo tanto la información de una página puede ser falsa, inválida, escrita incorrectamente o presentar otros errores.

1.4 Sistemas de Recuperación de Información

Los SRI según (BAEZA-YATES 2005), "*deben de alguna manera interpretar el contenido de la información dentro de una colección de documentos y establecer con ellos un orden de acuerdo al grado de relevancia que estos posean para las consultas de los usuarios"*.

1.4.1 Arquitectura de un Sistema de Recuperación de Información

Un SRI se define como el *"proceso donde se accede a una información previamente almacenada, mediante herramientas informáticas que permiten establecer ecuaciones de búsquedas específicas. Dicha información ha debido ser estructurada antes de su almacenamiento"*. (Pinto 2004)

"*Generalmente, todos los sistemas de recuperación de información comparten una misma arquitectura, 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 interfaz web (la más común), una interfaz de escritorio o ambas.*

[7] **JPEG:** (del inglés *Joint Photographic Experts Group*, Grupo Conjunto de Expertos en Fotografía), formato de compresión de imágenes

[8] **GIF:** *(del inglés Graphics Interchange Format)* es un formato gráfico utilizado ampliamente en la WWW, tanto para imágenes como para animaciones.

- ***Sistema de formulación de consultas:*** *realiza un pre-procesamiento de las consultas, trasladando las consultas hechas en 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 pre-procesada, para obtener un subconjunto de documentos relevantes que satisfagan la consulta introducida por el usuario, ordenados de acuerdo a un criterio de relevancia".* (López Herrera 2006)

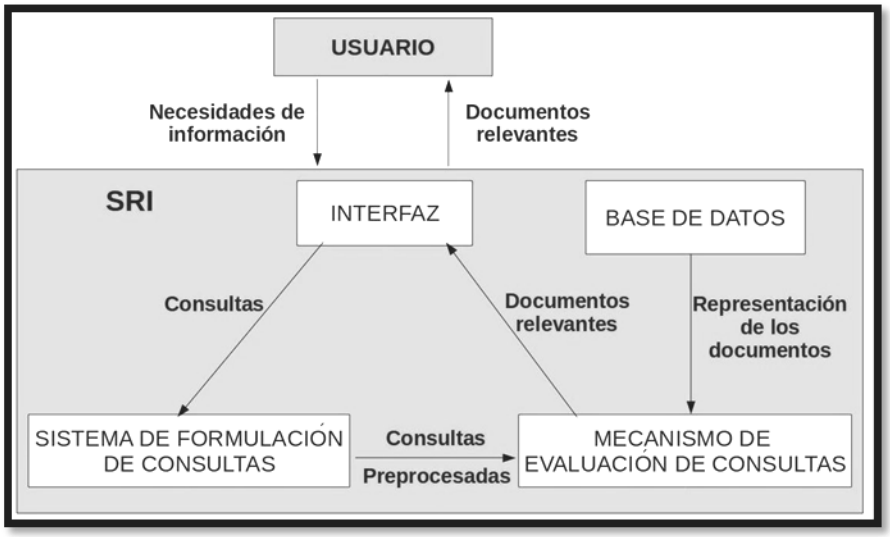


Figura 2. Arquitectura de un Sistema de Recuperación de Información (tomado de: Orión, un motor de búsquedas para la *Web* de la UCI).

1.4.2 Sistemas de Recuperación de Información

Actualmente existen dos formas básicas de recuperación de información en la *Web*:

- Búsqueda.
- Navegación.

Entre las principales herramientas utilizadas en Internet para recuperar información, se encuentran las siguientes:

- Buscadores o Motores de búsqueda (búsqueda).
- Meta-buscadores (búsqueda).
- Directorios (navegación).
- Buscadores mixtos (ambas).

1.4.2.1 Buscadores

"Los motores de búsqueda o buscadores web, son las herramientas de acceso a la información más populares y útiles en Internet, programados para la localización y recuperación de información en la red" (BOJO 2004). Los buscadores extraen de los documentos las palabras o términos que mejor los representen a través de un proceso automático, con estas palabras generan un índice invertido (ver **Figura 3.**) y almacenan esta información en una base de datos que puede ser interrogada por los usuarios a través de una interfaz.

Cuando los usuarios efectúan una consulta, el motor busca en la base de datos y devuelve, como respuesta, una lista con las *URLs* de aquellos documentos que se ajustan a los criterios establecidos en dicha consulta. Esta búsqueda no se efectúa directamente sobre la *Web* debido a su inmenso volumen y al tiempo requerido para dar respuesta a los usuarios. Estos sistemas consideran a la *web* como una extensa base de datos, lo que conlleva a que los mismos deban enfrentar algunos obstáculos que imponen las propias características presentes en la *web*.

En (BERGMAN 2001) el autor expresa que "*el 85% de los usuarios de Internet utilizan motores de búsqueda para satisfacer sus necesidades de información*. Los buscadores *web*, según Gonzalo Navarro en (GUTIÉRREZ 2008), "*permiten escribir un par de palabras y obtener inmediatamente abundante información, en general muy relevante, que satisface las necesidades de los usuarios".*

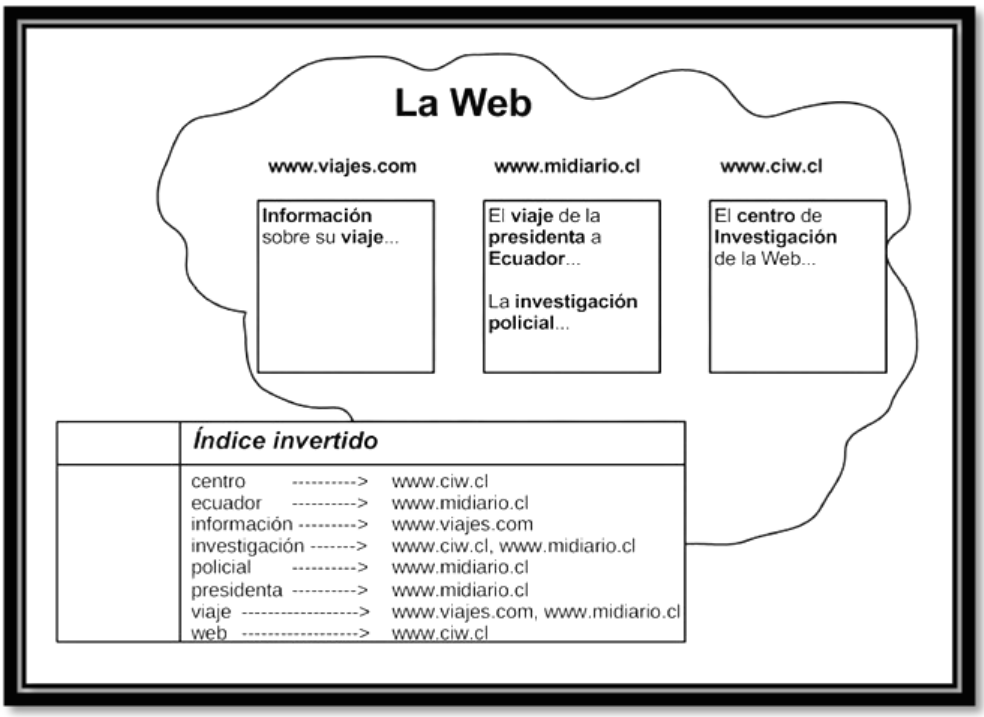


Figura 3. Ejemplo de índice invertido para tres páginas *web*. Tomado de (GUTIÉRREZ et al., 2008).

Sin un buscador los usuarios deberían conocer las direcciones en la *Web* de todos los sitios que pudieran resultarles interesantes o necesarios, perdiendo toda la información de los sitios que no conozcan. Los buscadores conectan la *Web*, pues existen grandes porciones de esta a las que solo se llega utilizando un buscador. Según (GUTIÉRREZ 2008), "*casi un tercio del tiempo que los usuarios pasan en Internet lo dedican a realizar búsquedas en la Web".*

En (BAEZA-YATES 2005) se expresa que "*desde la perspectiva del usuario existen dos requisitos principales que deben cumplir todos los buscadores: (1) un tiempo corto de respuesta y (2) 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".*

En (CASTILLO 2004) se explica que "*el diseño típico de un buscador web es en cascada (ver* ***Figura 4****), en el cual las operaciones son ejecutadas en el siguiente orden:*

- *Recolección de contenido web.*
- *Indexación del contenido recolectado.*
- *Búsqueda de información por parte de los usuarios.*

Se considera que el proceso de recolección es la etapa primaria y más importante dentro de la búsqueda en la Web.

Como método para recuperar información, los buscadores web presentan las siguientes ventajas:

- *Realizan el descubrimiento del contenido de la Web mediante un proceso automático, lo que permite analizar una mayor cantidad de información.*
- *Mantienen sus bases de datos más actualizadas que los directorios.*
- *Asignan a la información recolectada un orden de acuerdo con la calidad de su contenido.*
- *Permiten la realización de búsquedas avanzadas.*

Como desventajas se pueden señalar:

- *La clasificación de los contenidos es automática, por lo que carece de calidad en comparación con la de un directorio.*
- *La información encontrada no siempre responde a las necesidades reales de los usuarios. En ocasiones devuelven demasiadas URLs para una misma consulta, saturando de información al usuario. Esto provoca que se necesite examinar una amplia colección de documentos o formular una nueva consulta."*

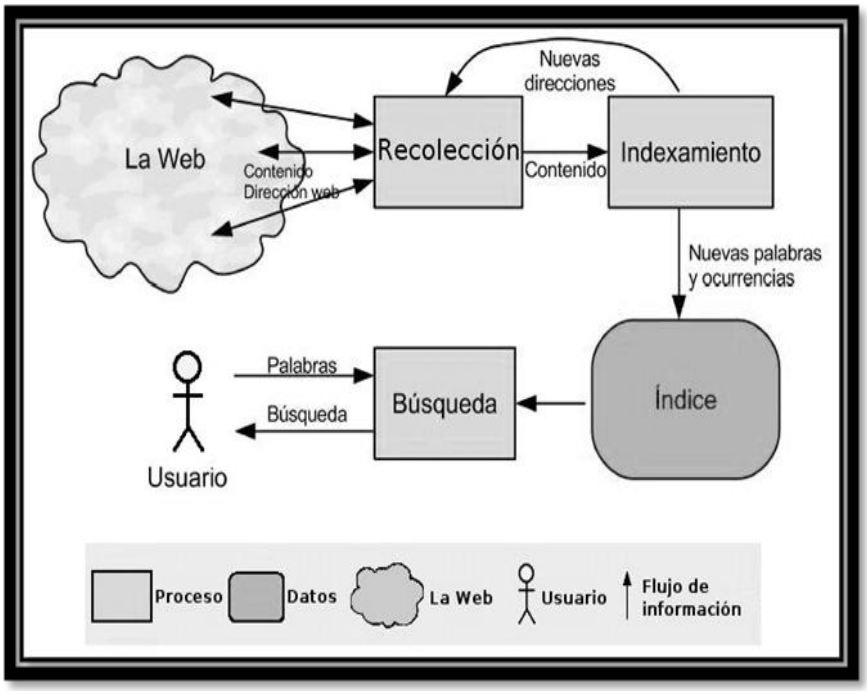


Figura 4. Diseño en cascada de un buscador *web*. Tomado de (GUTIÉRREZ 2008).

1.4.2.1.1 Arquitectura general de un motor de búsqueda

"La mayoría de los buscadores emplean una arquitectura araña-indexador centralizada, es decir, la araña (spider) y el componente de indización o indexación se encuentran unidos, (ver ***Figura 5****).*

La araña es la encargada de realizar peticiones a servidores distantes en busca de la información contenida en los mismos. Estas han evolucionado a un punto tal que permiten realizar peticiones por distintos protocolos de la familia TCP/IP[9] tales como: HTTP, FTP[10] entre otros.

Por su parte, el componente encargado de la indización, recibe las páginas recuperadas por la araña, extrae una representación interna de la misma y la almacena en forma de índices en una base de datos". (Aguirre 2007). "*Muchos indexadores emplean técnicas avanzadas para la extracción de vocabulario tales como:*

- ***Lista de palabras de parada o stopwords:*** *son listas de palabras muy habituales que no aportan significado a la información, por ejemplo, las preposiciones y los artículos.*
- ***Extracción de raíces o stemming:*** *consiste en extraer la raíz de las palabras con significado parecido, por ejemplo, plurales, tiempos verbales y otras".* (Aguirre 2007)

[9] **TCP/IP:** *Transmission Control Protocol, Internet Protocol.* (Protocolo de Transmisión y Control), (Protocolo de *Internet*).

[10] **FTP:** *File Transfer Protocol.* (Protocolo de Transferencia de Archivos).

"Todas estas palabras obtenidas por el indizador, son almacenadas en una base de datos o archivo invertido en forma de índices, lo que facilita la recuperación de la información por el motor de búsqueda.

El motor de búsqueda, por su parte, recibe la consulta de un usuario, que consiste en la introducción de un grupo de palabras claves sobre la información deseada. Estas palabras claves son convertidas por el sistema de formulación de consultas en un conjunto de incógnitas entendibles por el sistema, las que serán utilizadas por el subsistema de evaluación para devolver los documentos existentes en la base de datos, otorgando un orden de relevancia a dichos documentos en correspondencia con la consulta introducida por el usuario" (López Herrera 2006).

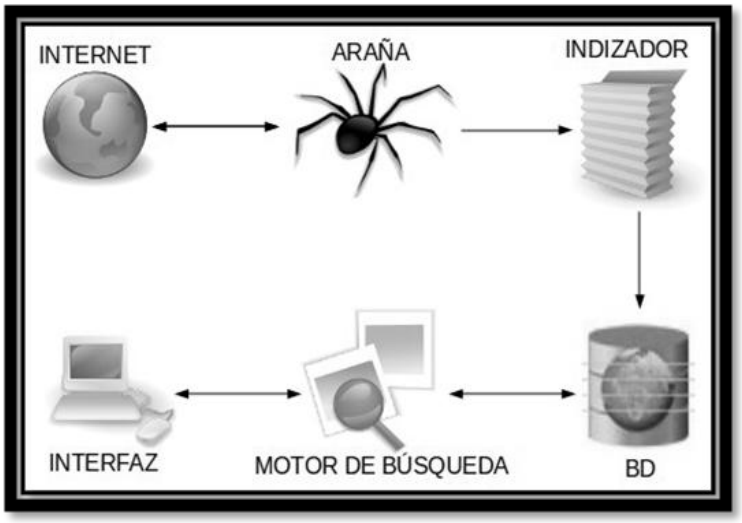


Figura 5. Arquitectura de un motor de búsqueda.

1.4.2.2 Meta-buscadores

"En realidad, no son buscadores, lo que hacen, *es realizar búsquedas en auténticos buscadores, analizan los resultados de la página y presentan sus propios resultados. No disponen de estructuras de indexación propias, reenvían las búsquedas a otros sistemas de búsqueda. Su único trabajo consiste en combinar las mejores páginas que ha devuelto cada buscador, logrando así un mayor abanico de resultados con mucha mayor calidad".*(Consoft 2007)

"Cada buscador utiliza su propia estrategia a la hora de recoger información de una página y a la hora de ordenar los resultados de las búsquedas, esto repercute en que las páginas de mayor relevancia en un buscador no tienen que coincidir en los del resto, aportando puntos de vista distintos.

Entre las principales ventajas de este sistema de búsqueda se encuentran:

- *Combina resultados de múltiples buscadores.*
- *Elimina los resultados duplicados.*

- *Confiere mayor peso a aquellos resultados que aparecen en varios buscadores.*

Como sus principales desventajas se pueden encontrar:

- *Es mayor el tiempo de espera.*
- *Explota los servicios ofrecidos por otros sistemas de búsqueda".* (Consoft 2007)

1.4.2.3 Directorios

"Los directorios o índices temáticos, son herramientas que organizan las páginas web jerárquicamente, o sea, permiten organizar la web por temas, lo que facilita la búsqueda de la información existente en un área determinada del conocimiento. Los resultados son recorridos en profundidad, lo que garantiza que al final de la jerarquía, exista una alta probabilidad de encontrar lo que realmente se necesita. Además, estos sistemas no poseen una araña u otro mecanismo automático que recorra la Web en busca de nueva información como suele suceder con los motores de búsqueda, sino que es operado por humanos". (Kiva 2009)

1.5 Comparación entre buscadores y directorios

Existen actualmente dos formas principales de indexar la información de Internet:

1. **Indexación manual:** es el mecanismo empleado en los sistemas de búsqueda con estructura de directorio. En este caso son los propios *webmaster*[11] o los responsables del sistema los encargados de llevar a cabo la indexación de documentos de forma manual bajo alguna de las categorías previamente establecidas en el directorio. Entre sus problemas se encuentra la lentitud en el análisis, actualización de la información y la confusión clasificatoria. Esto último se refiere a que la mayoría de los directorios no tienden a usar ningún sistema de clasificación establecido, sino que los propios responsables son los encargados de establecer las materias, las cuales en ocasiones no son muy intuitivas sino que confunden al usuario.
2. **Indexación automática:** es el tipo de indexación en los sistemas tipo buscadores o motores de búsqueda. Se auxilian de un programa denominado "*spider*" que se encarga de ir moviéndose por todas las páginas *web*, simulando la confección de una gran telaraña e ir recolectando la información de los documentos por donde va pasando y almacenándola. Entre las principales ventajas se encuentra la rápida actualización de los cambios producidos en Internet, además, al indexar prácticamente todos los términos del documento, la búsqueda se puede llevar a cabo por estos. Como principal desventaja, señalar que la precisión en la identificación del documento no es tan exacta como la manual, además muchas páginas pueden atraer la

[11] **webmasters:** compañero(s) encargado(s) de administrar un sitio o un conjunto de sitios webs.

atención hacia sí repitiendo palabras muy solicitadas por los usuarios. Otra desventaja que poseen es que la indexación de este tipo solo reconoce textos puros y en ocasiones ficheros de otro formato.

Algunas diferencias entre buscadores y directorios se describen a continuación:

- **Cobertura de red:** los directorios cubren mucho menos documentos que los buscadores ya que el proceso automático es mucho más rápido.
- **Profundidad de clasificación:** los directorios clasifican los documentos bajo encabezamientos de materias amplias, mientras que los motores permiten clasificarlo por cada una de las palabras (no vacías[12]) que contiene el documento.
- **Actualidad de los enlaces:** los motores tienden a tener los enlaces más actualizados que los directorios.
- **Resultados de búsqueda:** generalmente se obtienen más resultados utilizando motores de búsqueda que directorios.

1.6 Buscadores existentes

Actualmente existen un grupo de herramientas que permiten realizar distintos tipos de búsqueda. En esta sección se darán a conocer algunos detalles correspondientes a las herramientas de búsquedas utilizadas a nivel internacional, nacionales y dentro de estos los desarrollados en la UCI.

1.6.1 Buscadores internacionales

Google

"Google al igual que otros buscadores su búsqueda se basa en hipertextos, analizando todo el contenido web y la posición de todos los términos en cada página, incorpora una búsqueda simple y otra avanzada, su interfaz es sencilla y ofrece varios servicios", como se muestran en la **Figura 6**.

[12] **Palabras no vacías o *stopword*:** término que se refiere a las palabras muy comunes en el lenguaje y que aparecen un gran número de veces en cualquier texto, ejemplo: artículos, preposiciones, conjunciones, etc.

Figura 6. Servicios de Google.

"A continuación se exponen varios servicios que brinda Google:

- *Buscador de imágenes. (http://images.google.com/).*
- *Buscador dentro de los grupos. Los grupos se encuentran organizados por categorías y allí se puede encontrar un foro de debate o un grupo sobre el tema que se busca. (http://groups.google.com/).*
- *Buscador de noticias, en medios de comunicación de Internet (http://news.google-com/).*
- *Integración de una barra de búsqueda de Google dentro del navegador web. (http://www.google-com/tools/firefox/toolbar/FT5/intl/es/index.html).*
- *Traduce páginas web, palabras o textos en varios idiomas (inglés, español, alemán, francés, etc.). (http://www.google.com.cu/language_tools?hl=es).*
- *Servicio de correo utilizado por casi todos los internautas de la Red de Redes (http://www.gmail.com).*

Google se basa en la tecnología PageRank[13], para realizar el posicionamiento de las páginas web, lo que asegura que los resultados más importantes se muestran primero. Los complejos mecanismos automáticos de búsqueda de Google permiten prescindir de la interferencia humana. Está estructurado de manera que nadie puede comprar un lugar privilegiado en la lista ni alterar los resultados con fines comerciales (nadie puede comprar un PageRank más elevado, por ejemplo). Su robot de búsqueda es Googlebot desarrollado con el lenguaje Python[14]. Este colecciona documentos desde la WWW para construir una base de datos para el motor de búsqueda Google. Tiene dos versiones, Deepbot y Freshbot. Deepbot investiga profundamente, tratando de seguir cualquier enlace en esa página, además de poner esta página en el caché y dejarla disponible para Google. Freshbot investiga la Web buscando por contenido nuevo". (www.taringa.net 2009)

Yahoo

"Utiliza el motor de búsqueda "Search Assist" (asistente de búsqueda), incluyendo una función que permite ver sugerencias mientras el usuario se decide entre las opciones encontradas por el buscador. También permite incluir varios tipos de búsquedas en una misma página, combinando textos, fotos, videos o audio.

Yahoo Slurp es el robot rastreador o spider de Yahoo, el mismo recopila documentos de la WWW para construir un índice rastreable para servicios de búsqueda que usan el motor de búsqueda de Yahoo. Como parte del sistema de rastreo, Yahoo Slurp toma en cuenta los estándares robots.txt para asegurarse de que no se rastrean e indexan las páginas que no se quiere que aparezcan en resultados de búsqueda a través de Yahoo Search Technology. Si una página está protegida por un fichero robot.txt no será considerada para inclusión ni indexación en la base de datos de Yahoo. Yahoo Slurp almacena toda la información que recoge durante el proceso de rastreo y luego la pasa a la base de datos.

Diariamente rastrea la Web para mantener sus páginas actualizadas y dos veces por semana se encamina en busca de nuevos contenidos. Esta herramienta también presta servicios adicionales al cliente al igual que Google". (Comin 2001)

---

[13] ***PageRank*:** familia de algoritmos utilizados para asignar de forma numérica la relevancia de las páginas *web* o documentos indexados por un motor de búsqueda.

[14] ***Python*:** se utiliza como lenguaje de programación interpretado, ahorrando un tiempo considerable en el desarrollo de programas, pues no es necesario compilar.

Live Search

"Incluye servicios como, correo, mensajería, noticias y búsquedas de imágenes. *Este buscador fue lanzado por la compañía de Microsoft y previamente dependía de otros para listar sus búsquedas. En 2004 debuta una versión beta con sus propios resultados, impulsada por su propio robot (llamado MSNBot).*

MSNBot tiene un funcionamiento similar a GoogleBot. Rastrea la Web a través de los enlaces establecidos de una página a otra y a partir de esta información realiza una clasificación de los enlaces recibidos y la importancia de las páginas que los enlazan.

Esta herramienta está inspirada en el funcionamiento de Google, brinda la facilidad al usuario de cambiar de idioma (tiene 41 idiomas para escoger), una búsqueda simple y otra avanzada, en el caso de Live Search también utiliza los símbolos:

- ***+:** encuentra páginas web que contengan todos los términos que van precedidos por el símbolo +. Además, permite incluir términos que normalmente se omiten.*
- ***"":** encuentra las palabras exactas de una frase.*
- ***():** encuentra o excluye páginas web que contengan un grupo de palabras.*
- ***AND o &:** encuentra páginas web que contengan todos los términos o frases.*
- ***NOT o -:** excluye aquellas páginas web que contengan un término o frase.*
- ***OR o |:** encuentra páginas web que contengan algunos de los términos o frases".*(Big 2009)

Altavista

"Presenta una búsqueda simple y una avanzada, es capaz de realizar una búsqueda en lenguaje natural pues puede reconocer cuales de las palabras que introduce el usuario son de un real significado y es capaz de suprimir las palabras sin significado como: artículos, preposiciones, adverbios entre otras. Brinda un servicio de ayuda al usuario donde se encuentra información sobre los distintos tipos de búsqueda que se realizan. Ofrece servicios de noticias, directorio, traducción de página, además el usuario puede descargar una barra de herramientas para su navegador. Este buscador al pertenecer a las empresas Overture compradas por Yahoo, algunos de sus servicios re-direccionan a páginas de Yahoo.

Los resultados de la búsqueda pueden ser mostrados por un orden específico o ranking, basándose en los siguientes factores:

- *Las páginas largas con mucho texto significativo.*
- *Páginas con un buen sistema de navegación, con varios vínculos a páginas con contenido relacionado.*

- *La conectividad de las páginas, incluyendo no sólo cuantos vínculos hay hacia una página sino también desde dónde vienen los vínculos.*
- *El nivel de directorio donde se encuentra la página. Los más altos son considerados como más importantes. Si una página está muy al fondo, el spider no irá tan abajo y nunca la encontrará. Estos factores estáticos son re-calculados una vez a la semana, y según vaya mejorando la página irá subiendo en el ranking.*

El índice de AltaVista se construye enviando spiders (programas robot) que capturan texto y lo almacenan. En este proceso no interviene ninguna acción humana ni juicio. El principal spider, Scooter, tiene otros spiders que lo ayudan a realizar tareas específicas para ayudar a mantener el índice actualizado, cómo, por ejemplo, comprobar vínculos rotos - páginas que se han movido o borrado y no serán indexadas. En la búsqueda avanzada, permite especificar fechas, idioma y muestra en primer lugar frases o palabras que se indiquen en las opciones especificadas. Al realizar la búsqueda muestra el título y el primer párrafo de la página, la URL y la posibilidad de traducción, ofrece términos relacionados y otras páginas que pertenecen a la misma URL. Al final de la página sugiere fuentes alternativas de búsqueda donde se pone de manifiesto la retroalimentación (en algunos casos son enlaces que llevan a páginas de Yahoo para realizar las búsquedas desde allí).

Presenta además la posibilidad de elegir el idioma del buscador (despliega una ventana con 25 idiomas), se pueden realizar búsquedas de imágenes, audio y video, además de una búsqueda por campos". (www.posicionamiento-web.org 2009)

1.6.2 Buscadores nacionales

Buscador Cubano 2x3

"Primer buscador cubano en Internet. La función principal de esta herramienta es facilitar la búsqueda, revisión y consulta de los más diversos contenidos publicados en páginas y sitios web de la red de Cuba. El mismo contiene indexadas en su base de datos más de 100 mil direcciones de sitios cubanos en Internet correspondientes al dominio definido por la nación (.cu). El robot de búsqueda indexa diariamente aquellos sitios que poseen mayor nivel de actualización, como son los medios de prensa; el resto de los sitios se revisan con menor frecuencia proporcionando una actualización semanal a su base de datos.

El sistema cuenta con la opción de introducir manualmente nuevos sitios por los usuarios para luego ser recorridos por su robot, aunque para ser incorporados tienen que cumplir con ciertos criterios, primero que el sitio esté en funcionamiento, que esté en el dominio .cu y que cumpla con las condiciones de pertenecer a entidades de cubanas o mixtas basadas en el territorio nacional.

Entre las principales funcionalidades que ofrece es la búsqueda por palabras o frases. En este sentido el sistema no busca por todas las palabras introducidas, pues el mismo posee un listado de palabras tales como preposiciones, conjunciones, artículos y otras que no aportan significado por sí mismas y que son conocidas como palabras vacías o stopwords. También permite realizar búsquedas especiales en sitios de medios de prensa así como en los discursos del compañero Fidel. El sistema ofrece la búsqueda de imágenes, en la misma se puede seleccionar diferentes tamaños para las imágenes. El criterio de ordenamiento para devolver los resultados está dado en el siguiente orden:

1. *Páginas que contienen las palabras tecleadas en la dirección URL.*
2. *Páginas que contienen las palabras tecleadas en el título.*
3. *Páginas que contienen las palabras tecleadas en los metadatos.*
4. *Páginas que contienen las palabras tecleadas en el cuerpo o contenido principal de la página.*

En la portada se presentan categorías o temas que agrupan la información disponible, cuenta con una búsqueda avanzada, sencilla y de fácil comprensión para los usuarios, pero carente de algunas opciones. Además cuenta con varios servicios de búsqueda especializada como la de imágenes, prensa, discursos de Fidel, archivo (Word y PDF), multimedia y más (tiempo, noticias, diccionarios). Uno de los principales problemas que presenta es que en los primeros resultados de una búsqueda muestra noticias muy desactualizadas".(bvs.sld.cu 2007)

1.6.3 Buscadores en la UCI

Faro fue desarrollado por un grupo de estudiantes y profesores de la antigua facultad 10 de la UCI en el año 2008. Para la recolección de las páginas *web*, utiliza un *spider* llamado *Wire* de origen chileno y para la creación de los índices un indexador de nombre *Swish-e* en su versión 2.4.5. Actualmente el proyecto está descontinuado debido a la falta de personal necesario para el desarrollo del mismo.

También han existido otros intentos de creación de un buscador *web* para la UCI, tal es el caso de Gugle. Este es una parodia del motor de búsqueda Google. Surge en el año 2008 como respuesta a una creciente necesidad de buscar y encontrar información útil en los sitios existentes en la universidad, hoy día el proyecto está descontinuado. Otro caso es Orion**,** fue desarrollado para la recolección de las páginas *web* en la universidad, utiliza el *spider* contenido en *mnoGoSearch*[15] para la implementación del motor de búsqueda, actualmente el proyecto está desactualizado.

[15] ***mnoGoSearch*:** motor de búsqueda *web*, compuesto por un indexador y un *front-end* (compilador) *web* para realizar búsquedas sobre el contenido indexado.

1.7 Interfaz de programación de aplicaciones

"Las siglas API provienen del inglés Application Programming Interface (Interfaz de Programación de Aplicaciones). Estas constituyen un conjunto de funciones y procedimientos o métodos, en la programación. Son usadas generalmente en las bibliotecas, denominadas comúnmente librerías para ser utilizadas por otro software como una capa de abstracción. Su propósito principal consiste en brindar un conjunto de funciones de uso general, evitando a los programadores tener que desarrollar todo un software desde el principio".(Stoughton 2005)

1.8 Proceso, métodos, metodología, herramientas y tecnologías a utilizar

A continuación se realiza un estudio de las herramientas, métodos y tecnologías que pueden ser utilizadas en el desarrollo del buscador de la Red Social de la Universidad de las Ciencias Informáticas.

1.8.1 Proceso de desarrollo del *software*

Proceso desarrollo de *software*

"Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto que reúne los requisitos del cliente. "Define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" mediante la definición de actividades, artefactos y roles". Un proceso de software detallado y completo suele denominarse "Metodología".

El proceso de desarrollo de software no es único. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo. Las metodologías de desarrollo pueden clasificarse en ágiles y pesadas o tradicionales".

Enfoque ágil

"El enfoque ágil de las metodologías de desarrollo de software surge como alternativa a las metodologías formales, que se consideraban excesivamente pesadas y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo. Dicho enfoque se basa:

- *Valorar más a los individuos y su interacción que a los procesos y las herramientas.*
- *Valorar más el software que funciona que la documentación exhaustiva.*
- *Valorar más la colaboración con el cliente que la negociación contractual.*
- *Valorar más la respuesta al cambio que el seguimiento de un plan.*
- *Valorar más el software que funciona que la documentación excesiva. Genera la documentación necesaria del desarrollo del producto."*

CMMI

"CMMI por sus siglas en inglés Capability Maturity Model Integration es un modelo de calidad del software que provee lineamientos para las empresas, organizaciones o áreas que deseen una mejora continua y efectiva en sus procesos de desarrollo. Para esto propone 25 áreas de proceso las cuales están agrupadas en 5 niveles (iniciado, gestionado, definido, gestionado cuantitativamente, en optimización) de madurez para clasificar a las empresas en función de qué áreas de proceso consiguen sus objetivos y se gestionan con principios de ingeniería.

El nivel 2 de CMMI es el gestionado, se enfoca en la gestión de procesos y define siete áreas:

Administración de requisitos: *mantiene los requisitos y describe las actividades para obtener y controlar los cambios a los requisitos y asegura que otros planes y datos relevantes se mantengan actualizados.*

Planificación del proyecto: *incluye el desarrollo del plan de proyecto, la involucración de las partes interesadas de forma apropiada, la obtención de compromisos con el plan y el mantenimiento del plan.*

Seguimiento y control del proyecto: *incluye las actividades de monitorización y toma de acciones correctivas. Estas acciones pueden incluir la re-planificación del proyecto.*

Gestión de acuerdos con proveedores: *trata la necesidad del proyecto de adquirir aquellas partes del trabajo que son producidas por proveedores. Las fuentes de productos que pueden ser usadas para satisfacer los requisitos del proyecto se identifican de forma proactiva.*

Medición y análisis: *da soporte a todas las áreas de proceso, proporcionando prácticas específicas que guían a los proyectos y a las organizaciones durante la alineación de las necesidades y objetivos de medición con una forma de medir que proporcionará resultados objetivos.*

Aseguramiento de calidad de procesos y productos: *da soporte a todas las áreas de proceso, proporcionando prácticas específicas para evaluar objetivamente los procesos, los productos de trabajo y los servicios realizados frente a las descripciones aplicables de procesos, estándares y procedimientos.*

Gestión de configuración: *brinda soporte a todas las áreas de proceso, estableciendo y manteniendo la integridad de los productos de trabajo usando la identificación de la configuración, el control de la configuración, los informes de estado de la configuración y las auditorías de la configuración".*(Estrada 2012)

A partir de lo explicado anterior se puede decir que el proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI reúne:

- Las mejores prácticas del modelo traducidas en metas que indican qué hacer y que guían la gestión del proceso.
- Las buenas prácticas del proceso de desarrollo ágil traducidas en tareas de ingeniería enfocadas en cómo hacer las cosas.

1.8.2 Herramienta de modelado

A continuación se describe la herramienta de modelado Visual Paradigm.

Visual Paradigm

"Herramienta CASE[16] que provee soporte al modelado visual con UML[17]. Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación".(Paradigm 2012)

1.8.3 Métodos de comunicación entre aplicaciones

***Representational State Transfer* (REST)**

"La Transferencia de Estado Representacional define una colección de principios arquitectónicos por los cuales se diseñan servicios web haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia clientes escritos en diversos lenguajes. Estos principios arquitectónicos resumen como los recursos son definidos y diseccionados. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos sobre HTTP sin una capa adicional, como hace SOAP.

Este mecanismo se centra en el acceso a los recursos nombrados a través de una única interfaz. Ofrece verdaderos servicios web sobre la base de URL y HTTP como protocolos de transportes. Utiliza comandos de HTTP para acceder a los datos. El desarrollo de las API y la documentación es mucho

[16] **CASE:** del inglés (*Computer Asisted Software Engeneering*), Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software*.

[17] **UML:** del inglés (*Unified Modeling Language*) o Lenguaje de Modelación Unificado.

más fácil de entender. Permite un mejor soporte para los clientes. Además es compatible con SSL[18]*. Tiene un gran rendimiento y escalabilidad, es ligero debido a que no hace falta XML*[19] *de configuración. Ofrece resultados legibles y es fácil de implementar ya que no hacen falta herramientas específicas".* (Massé 2012)

Simple Object Access Protocol (SOAP)

"Es un protocolo que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma y del lenguaje. Define, cómo dos objetos en diferentes procesos se comunican por medio de intercambio de datos XML.

Basado en lecturas, no se puede almacenar en caché. Además sólo permite formato de datos XML. Se centra en la exposición de piezas de lógica de aplicación (no de datos) como los servicios. Además de ser compatible con SSL, también es compatible con el protocolo de comunicaciones que suministra un medio para aplicar seguridad a los servicios web WS-Security. Proporciona una implementación estándar de integridad y privacidad de los datos. Al contrario de REST no posee un sistema de mensajería estándar y espera a sus clientes para hacer frente a fallas en la comunicación, SOAP tiene éxito, ya que tiene lógica de reintento integrado". (Lazo and Reyes 2007)

1.8.4 Servidor *web*

Un servidor *web* es un programa que implementa el protocolo HTTP. Este protocolo está diseñado para transferir hipertextos, páginas *web* o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. Este servidor se encarga de mantenerse en espera de las peticiones HTTP llevadas a cabo por un cliente, donde este realiza una petición al servidor, quien le responde con el contenido solicitado.

Apache

"Apache es un servidor web flexible, rápido, eficiente y continuamente actualizado. Entre sus características se destacan:

- *Es multi-plataforma.*
- *Es un servidor web conforme al protocolo HTTP.*

[18] ***SSL:*** del inglés (*Secure Sockets Layer*), la capa de conexión segura, es un protocolo criptográfico que proporciona comunicaciones seguras por una red.

[19] **XML:** del inglés (*eXtensible Markup Language*). Es una meta-lenguaje que permite definir lenguajes de marcado.

- *Es modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con la API de programación de módulos.*
- *Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.*
- *Se desarrolla de forma abierta.*
- *Es extensible.*
- *Es popular (fácil de conseguir ayuda y soporte)".*(Bannister 2012)

1.8.5 Lenguajes de programación

Los lenguajes de programación para la w*eb* se clasifican en lenguajes del lado del cliente y del lado del servidor.

1.8.5.1 Lenguajes y tecnologías utilizadas del lado del cliente

HTML

"Lenguaje que se utiliza para crear las páginas web. Indica a los navegadores cómo deben mostrar el contenido.

Este lenguaje contiene dos partes:

1. *El contenido, que es el texto que se verá en la pantalla de un ordenador.*
2. *Las etiquetas y atributos que estructuran el texto de la página web en encabezados, párrafos, listas, enlaces, etc. y normalmente no se muestra en pantalla".* (Masadelante.com)

CSS

*"Hojas de Estilo en Cascada**.** Es un lenguaje del lado del cliente que permite, una vez creados los contenidos mediante el lenguaje HTML/XHTML[20], definir la manera en la que se va a mostrar la presentación de los documentos electrónicos, es decir, es utilizado para diseñar, modificar, especificar y describir mediante las hojas de estilo como será visualizado el contenido de la página. Por lo tanto, mediante este lenguaje es posible separar el estilo visual del contenido".*(Pérez 2008)

JavaScript

"Es un lenguaje de programación interpretado que ha permitido un gran desarrollo en la animación de las páginas web. Está basado directamente en objetos y guiado por eventos. Su diseño se enfoca

[20] **XHTML:** Lenguaje de Marcado de Hipertexto Extensible de su significado en *inglés eXtensible HyperText Markup Language.*

específicamente para el desarrollo de aplicaciones cliente-servidor, encargándose de efectuar acciones del lado del cliente".(Pérez 2008)

1.8.5.2 Lenguaje utilizado del lado del servidor

Estos lenguajes son necesarios para desarrollar la lógica de negocio en el servidor, acceder a las bases de datos y procesar la información.

PHP

"Lenguaje script de alto nivel interpretado del lado del servidor. Permite acceder a los recursos con los que cuenta el servidor. No es necesario ser soportado por el navegador, no obstante, para que funcione debe ser soportado por el servidor. Debido a su código abierto, es utilizado por una gran cantidad de desarrolladores, garantizando así que los fallos sean resueltos con rapidez, por lo tanto, el código es continuamente mejorado. Puede interactuar con muchos motores de bases de datos como: MySQL[21], PostgreSQL[22], Oracle[23], entre otros. Está integrado a varias bibliotecas externas para el manejo de los datos, la generación de documentos en PDF, calendarios, XML, entre otros. Permite técnicas de programación orientada a objetos".(Group 2009)

1.8.6 Marco de trabajo

Restler

Es un marco de trabajo único y completo basado en el lenguaje de programación PHP. Permite a los clientes enviar datos en cualquiera de los formatos habilitados o como una cadena de consulta o formulario de envío, que convierte los datos y asigna los parámetros de la función. Soporta los métodos de petición de HTTP, además deja al desarrollador la opción de solo asignar las funciones al método URL. Es gratuito y de código abierto bajo licencia GNU. Posee métodos públicos y protegidos que gestionan sus propias dependencias. Además permite agregar seguridad a las API desarrolladas ya que posee esquemas de autenticación conectables.

JQuery

"JQuery es un marco de trabajo JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar

[21] **MySQL:** sitio oficial, http://www.mysql.com.

[22] **PostgreSQL:** sitio oficial, http://www.postgresql-es.org.

[23] **ORACLE:** sitio oficial, http://www.oracle.com.

interacción con la tecnología AJAX a páginas web. Es un producto estable, bien documentado y con un gran equipo de desarrolladores a cargo de su mejora". (Álvarez 2012)

1.8.7 Entorno integrados de desarrollo

*"Un Entorno Integrado de Desarrollo (IDE de su significado en inglés **I**ntegrated **D**evelopment **E**nvironment) es un grupo o colección de programas necesarios que han sido creados como un programa de aplicación, por lo tanto, es la unión de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica, suministrando un marco de trabajo para la mayoría de los lenguajes de programación y en ocasiones puede funcionar también como un sistema en tiempo de ejecución, utilizando el lenguaje de programación de forma interactiva".(Herrera 2007)*

Entre los entornos integrados de desarrollo se encuentran:

Netbeans

"NetBeans es un entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto de NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como PHP, JavaScript y Ajax, Ruby y Ruby, Groovy y C/C++. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos, así como una selección diversa de complementos de tercero". (Netbeans 2012)

Zen Studio

IDE que sirve de editor de texto para páginas PHP. Divide sus funcionalidades de la parte del cliente y las de la parte del servidor, instalándose cada una de ellas por separado. Permite realizar depuraciones sencillas de script.

1.9 Buscadores e indexadores

Nutch

"Nutch es un buscador web de código abierto basado en la librería Java Lucene. Actúa como araña para la recolección de la información existente en la Web. Permite, haciendo uso de analizadores de varios formatos de archivos, extraer información desde archivos PDF, XML, HTML y otros.

Es considerada la solución de código abierto más usada en motores de búsqueda. Posee una amplia comunidad de desarrolladores y usuarios. Su desarrollo está patrocinado por la Fundación Apache y Lucid Imagination." (Nutch 2009)

"Entre sus principales características se destacan:

- *Captura, indización en modo paralelo y distribuido.*
- *Extensible mediante plugins*[24]*.*
- *Soporte para diversos formatos, tales como: texto plano, HTML, XML, PDF; JS*[25]*, RSS*[26]*.*
- *Ontologías.*
- *Solución basada en clúster.*
- *Sistema de fichero distribuido.*
- *Soporte para autenticación NTLM*[27]*." (Nioche 2009)*

Swish-e

"Swish-e es un sistema rápido, flexible y de código abierto para la indexación de colecciones de páginas web y otros archivos.

Entre sus principales características destacan:

- *Soporte para archivos de texto plano, HTML, PDF, e-mail y XML.*
- *Soporte para el SGBD*[28] *MySQL.*
- *Fichero de índice portable entre plataformas.*
- *Soporte para búsqueda de frases.*
- *Desarrollado en el lenguaje C con algunos script en Perl.*
- *Posee un script CGI*[29] *para realizar búsquedas sobre los índices generados.*
- *Su licencia es GPL (General Public License) y posee versiones tanto para GNU*[30]*/Linux como para Windows.*
- *Posee una amplia comunidad aunque la documentación en idioma español es escasa".* (Swish 2010)

[24] ***Plugin*:** Es una aplicación que se relaciona con otra, para aportarle una función nueva y generalmente muy específica.

[25] **JS:** siglas para referirse al lenguaje *JavaScript.*

[26] **RSS:** del inglés, *Really Simple Syndication.*

[27] **NTLM:** protocolo de autenticación.

[28] ***SGBD*:** Sistema Gestor de Base de Datos.

[29] **CGI:** del inglés, *Common Gateway Interface* o Pasarela de *Interfaz* Común.

[30] **GNU:** del inglés, *Lesser General Public License.*

Solr

Es una popular herramienta de indexación de datos rastreados por un motor de búsqueda. Su desarrollo está patrocinado por la Fundación *Apache* y *Lucid Imagination.* Sus características principales incluyen una búsqueda de texto completo, resaltado de búsqueda de facetas, la integración de bases de datos y documentos (por ejemplo, *Word*, *PDF).* Solr es altamente escalable, proporcionando búsqueda distribuida y replicación de índices, es desarrollado en *Java* y se ejecuta como un servidor de búsqueda independiente de texto completo dentro de un contenedor de *servlets*. Solr utiliza la biblioteca de *Java Lucene* para la búsqueda e indexación de texto completo y tiene *APIs* que hacen que sea fácil de utilizar desde cualquier lenguaje de programación. Presenta una configuración externa de Solr, permitiendo adaptarse a casi cualquier tipo de aplicación sin necesidad de codificación en *Java*.

Htdig

Htdig es un sistema de indexación sobre la *Web* y motor de búsqueda para un dominio o una intranet. Este sistema no pretende sustituir los grandes sistemas de búsqueda en Internet tales como Google, Yahoo y otros, sin embargo, se destina a cubrir las necesidades de búsqueda en una empresa, escuela o sección dentro un sitio *web*. Su funcionamiento se basa en el protocolo *HTTP*. El proyecto se descontinuó en junio del año 2004. Fue desarrollado en C/C++. Su curva de aprendizaje es alta.
Entre sus principales características se destacan:

- Soporte para *HTML* y texto plano.
- Búsqueda de expresiones booleanas.
- Resultados de las búsquedas configurables.
- Soporte para la exclusión de *robot.txt.*
- Búsquedas en una sub-sección de una base de datos.
- Lanzado bajo la licencia *GNU*.

Metodología para evaluar buscadores

La metodología para evaluar el funcionamiento y la eficiencia de los SRI en la UCI, propuesta en un trabajo de diploma realizado en la propia universidad. Describe las pruebas mediantes métricas, agrupándolas en métricas técnicas orientadas a medir de manera concreta y precisa la efectividad del sistema y en métricas de calidad para evaluar la calidad de los resultados devueltos así como del proceso de recuperación de información

1.10 Fundamentación del proceso de desarrollo

El desarrollo de la aplicación se guiará por el proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI, la Universidad se encuentra en un arduo proceso de mejora, rigiendo como característica el uso de este proceso para guiar el desarrollo de aplicaciones dentro de la institución.

1.11 Fundamentación de las tecnologías, lenguajes y herramientas a utilizar

Durante el estudio de las tecnologías se trataron los aspectos más importantes de las mismas que ayudarían en la elección de las más convenientes para el desarrollo del buscador basado en palabras claves de la RSU de la UCI. Este sistema forma parte de los proyectos de la Universidad que se rigen por los lineamientos generales que son especificados por la Dirección del Centro de Informatización Universitaria, que se ocupa entre otras cosas, de estandarizar el empleo de tecnologías en los sistemas que son desarrollados por el este. La herramienta para el modelado del sistema que se utiliza es Visual Paradigm v8.0, que a pesar de no ser libre, la Universidad tiene la licencia y es una herramienta que por sus características, cubre todo el ciclo de vida de un proyecto y permite diseñar diagramas de clases, generar código desde diagramas y documentación. El lenguaje del lado del servidor que se emplea en el desarrollo de la propuesta de solución es PHP v5.3.10 por las ventajas que presenta para los desarrolladores *web* en cuanto a facilidad, flexibilidad y su abundante documentación. Para la confección y darles formato a las páginas se utilizan HTML v5 y CSS como lenguajes de marcado. Como servidor *Web* se utiliza Apache v 2.2 debido a su flexibilidad, rapidez, estabilidad, eficiencia y sus constantes actualizaciones, presenta soporte para el lenguaje PHP. Como IDE de desarrollo se empleó el NetBeans v7.0.1 que es un producto de código abierto que incluye estándares de la industria de *software* ampliamente aceptados, con excelente soporte para autocompletado de código PHP. Para la comunicación del API a desarrollar con la Red Social de la Universidad de las Ciencias Informáticas, se emplea REST por ser más eficaz, asequible y mejor rendimiento que SOAP. En el desarrollo del API se utiliza el marco de trabajo Restler v2.0.1 para la creación de la capa de servicios del buscador. *Solr v3.5* es el encargado de indexar los datos rastreados por un motor de búsqueda, en este caso se hará uso de *Nutch v1.4.*

1.12 Conclusiones parciales

En el capítulo concluido se realizó un breve análisis sobre los antecedentes de los buscadores existentes a nivel nacional e internacional, determinándose que los mismos no satisfacen las necesidades de la Red Social de la Universidad de las Ciencias Informáticas; debido a que no indexan la información contenida en la red de la universidad, además su uso requiere un gasto innecesario de cuota de Internet por parte de sus usuarios. Para el desarrollo de este sistema se estudiaron las

posibles tecnologías y herramientas a utilizar, comparando las mismas en cuanto a sus ventajas y desventajas para poder obtener un mayor dominio de su uso, seleccionando las que más se ajustan a la solución planteada.

Capítulo II. Propuesta de solución del buscador de la Red Social de la Universidad de las Ciencias Informáticas

2.1 Introducción

En el presente capítulo se realiza una descripción de la solución propuesta para el buscador de la Red Social de la Universidad de las Ciencias Informáticas. Se describen los principales requisitos funcionales y no funcionales relacionados con el funcionamiento del sistema. Se describe el estándar de codificación utilizado para una mejor legibilidad del código y se presenta además el modelo de despliegue y el modelo de dominio utilizado para la construcción de la solución propuesta.

2.2 Situación problemática y propuesta de solución

Las personas en la UCI no pueden realizar una búsqueda sobre toda la información publicada en la red, debido a que las herramientas existentes no se encuentran en condiciones favorables de uso por estar descontinuadas, desactualizadas o sin soporte, lo que provoca que en ocasiones se desconozcan los temas publicados en la red, no usándose la misma de forma satisfactoria para apoyar las temas de investigación en la universidad. Por tal razón, los usuarios de la institución hacen uso de los buscadores internacionales como (Google, Yahoo), para buscar información cuando en ocasiones resulta que lo que se desea encontrar está disponible dentro de la propia universidad.

La presente investigación tiene como objetivo principal la propuesta de una herramienta que permite la búsqueda de información como documentos e imágenes disponibles en la *web* de la universidad. De esta forma los usuarios de la institución podrán encontrar información de interés sin tener que consumir cuota de navegación por hacer uso de los motores de búsqueda internacionales.

Para el desarrollo del buscador se utilizan como herramienta de búsqueda a Nutch en su versión 1.4 y como rastreador de los sitios publicados de la universidad el indexador Solr en su versión 3.4.0, el cual se encarga de crear el índice de las URLs encontradas por el rastreador antes mencionado.

2.3 Análisis de lo solución propuesta

En esta sección se describe la propuesta del buscador de la Red Social de la Universidad de las Ciencias Informáticas.

2.3.1 Modelo de dominio

"El modelo de dominio es un artefacto que puede ser tomado como punto de partida para el diseño de un sistema, ocupando un rol protagónico en su desarrollo. Es una representación visual estática que contiene los conceptos relacionados con el dominio cuyo objetivo es ayudar a comprender los

elementos con los que trabajan y utilizarán los usuarios".(Jacobson, Boosh et al. 2000) Fue necesario utilizar modelo de dominio debido a la ausencia de procesos de negocio.

Para el sistema a desarrollar se define el siguiente diagrama de clases del dominio (ver **Figura 7.**):

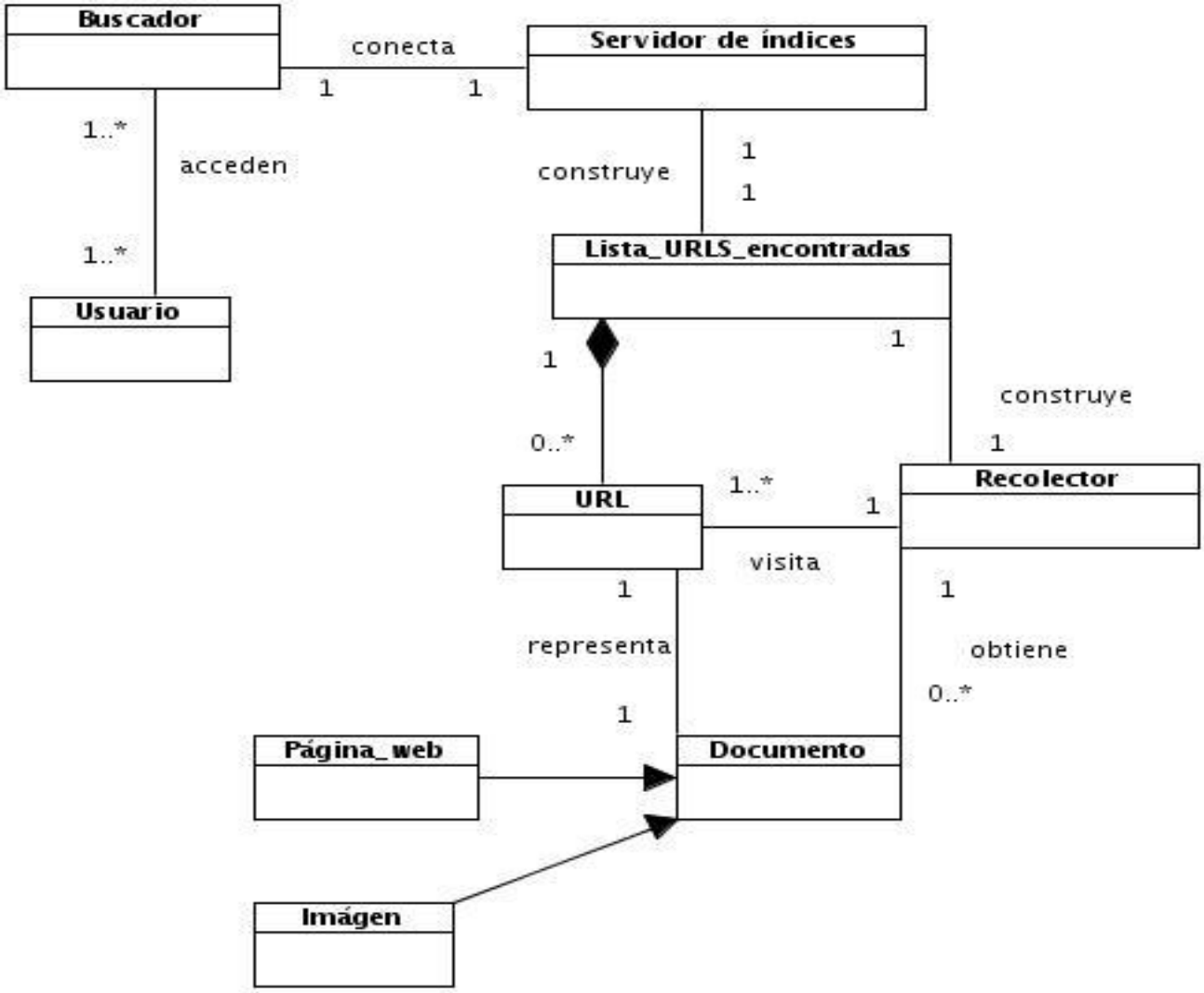


Figura 7. Modelo de dominio del buscador de la Red Social de la UCI.

2.3.1.1 Definición de las clases del dominio

URL: representa la dirección en la *Web* de un único documento.

Documento: archivo disponible en la *Web* identificado por una URL que puede representar, entre otros formatos, una imagen, un archivo multimedia, un archivo de texto o una página *web*.

Página *web*: archivo de texto codificado en lenguaje HTML, puede contener enlaces a otros documentos por medio de sus respectivas URLs.

Lista de URL encontradas: representan las URLs detectadas por el Recolector al extraer los enlaces de las páginas *web* que visita durante el recorrido.
Recolector: dada una lista de URLs de inicio comienza un recorrido en el que se detectan nuevas URLs, las cuales son agregadas a la lista de URLs encontradas para ser visitadas posteriormente.
Servidor de índices: dada la lista de URLs encontradas, crea un índice con los documentos encontrados al realizar la visita de dichas URLs.
Usuario: esta entidad representa a las personas que consumen las funcionalidades del sistema externo.
Buscador: constituye el resultado principal que se obtiene al concluir la investigación, un buscador para la Red Social de la UCI.

2.3.2 Requisitos de la propuesta de solución

"Los requisitos son condiciones o necesidades que posee un usuario para resolver un determinado problema o alcanzar un objetivo. Son declaraciones de los servicios que provee el sistema, definiendo la manera en que este reacciona a entradas particulares. Estos pueden dividirse en requisitos funcionales y requisitos no funcionales". (Sooft 2007)
Para la propuesta de solución se definieron los requisitos funcionales y no funcionales que se listan a continuación:

2.3.2.1 Requisitos funcionales

"Los requisitos funcionales definen las funciones que es capaz de realizar el sistema, es decir, describen las funcionalidades o los servicios que se espera que este provea".(Códova 2007) A continuación se muestran los requisitos funcionales del buscador de la Red Social de la UCI:

Tabla 1. Requisitos funcionales del buscador de la Red Social de la UCI.

No	Nombre	Descripción	Complejidad	Prioridad para el cliente
RF 1	Búsqueda básica	La búsqueda básica debe permitir la recuperación de aquellos recursos que se encuentran publicados en la red, partiendo de la consulta formulada por el usuario.	Alta	Alta
	Campo(s) (Entrada)	**Tipo de Datos**	**Reglas o Restricciones**	

	consulta	*varchar*	Solo letras, no se aceptan números o caracteres extraños. **Ejemplo:** nova.	
	Observaciones:			
No	**Nombre**	**Descripción**	**Complejidad**	**Prioridad para el cliente**
RF 2	Búsqueda de imágenes	Búsqueda por tipo de contenido. Debe recuperar todos los archivos con las extensiones de imágenes.	Alta	Alta
	Campo(s) (Entrada)	**Tipo de Datos**	**Reglas o Restricciones**	
	consulta	*varchar*	Solo letras, no se aceptan números o caracteres extraños. **Ejemplo:** nova.	
	Observaciones:	El sistema solo recupera aquellos resultados que coincidan con alguno de los archivos de imágenes más comunes, como son: .jpg, .gif, .png, .jpeg, entre otros.		
No	**Nombre**	**Descripción**	**Complejidad**	**Prioridad para el cliente**
RF 3	Búsqueda de documentos	Búsqueda por tipo de contenido. Debe recuperar todos los archivos con las extensiones de documentos.	Alta	Alta
	Campo(s) (Entrada)	**Tipo de Datos**	**Reglas o Restricciones**	
	consulta	*varchar*	Solo letras, no se aceptan números o caracteres extraños. **Ejemplo:** nova.	
	Observaciones:	En la búsqueda de solo documentos, el sistema intenta recuperar todos aquellos documentos, ya sean PDF, documentos de Microsoft Word, archivos de texto plano u otro tipo de documento conocido.		

2.3.2.2 Requisitos no funcionales

"Los requisitos no funcionales son aquellos que no se refieren directamente a las funciones que debe realizar el sistema, sino a las características que puedan limitar el sistema de una forma u otra".(Códova 2007)

Tabla 2. Requisitos no funcionales del buscador de la Red Social de la UCI.

Requisitos no funcionales.	
Usabilidad	**RNF 1.1:** el sistema debe presentar una interfaz que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. **RNF 1.2:** debe poseer una interfaz de manejo cómoda, que posibilite a los usuarios sin experiencia una rápida adaptación. **RNF 1.3:** debe ser compatible con los principales navegadores *web*. **RNF 1.4:** el uso de la interfaz de programación de aplicaciones brinda la posibilidad de ser consumida por cualquier aplicación.
Soporte	**RNF 2.1:** el API contará con un grupo de soporte y asesoría al cliente del producto destinado a brindar asesoría y soporte técnico al mismo. **RNF 2.2:** para garantizar el soporte de esta herramienta, se debe contar con un manual de ayuda para los usuarios.
Restricciones de diseño	**RNF 3.1:** lenguaje de programación PHP v5. **RNF 3.2:** servidor *web* Apache v2.2. **RNF 3.3:** entorno integrado de desarrollo (IDE) NetBeans v7.01. **RNF 3.4:** herramienta de modelado Visual Paradigm v8.4. **RNF3.5:** como proceso de desarrollo "Proceso de desarrollo con enfoque ágil orientado al segundo nivel de CMMI".
Documentación de usuarios en línea y ayuda del sistema	**RNF 4.1:** el buscador brinda como apoyo una Ayuda contextual en la cual se refleja detalladamente la explicación de cada una de las funcionalidades que brinda.
Componentes comprados	**RNF 5.1:** la licencia de Visual Paradigm fue adquirida por la universidad.
Interfaz	**RNF 6.1:** interfaz intuitiva, fácil de usar, sencilla, interactiva y debe mantener el mismo formato en todas las páginas. **RNF 6.2:** el diseño gráfico será acorde con las pautas de diseño del centro.

Hardware	**RNF 7.1:** debe existir una red de área local para consumir los servicios que brinda el buscador. **RNF 7.2: para explotación del servidor web:** memoria RAM de 4 GB, 1 Procesador Intel Dual Core o Core 2 Duo a 2.6 Ghz mínimo y 80 de disco duro. **RNF 7.3: para el desarrollo:** una PC de 1 GB de RAM y 80 GB de disco duro con Sistema Operativo Ubuntu 10.04 o superior. **RNF 7.4: para explotación del cliente:** una PC Pentium 3 o superior, CPU 133 MHZ o superior, 256 RAM mínimo, 512 RAM recomendada o superior. **RNF 7.5: para explotación del servidor de índice:** memoria RAM de 4 GB, 1 Procesador Intel Dual Core o Core 2 Duo a 2.6 Ghz mínimo y 160 de disco duro.
Software	**RNF 8.1:** servidor *web*: Apache v2.2. **RNF 8.2:** las estaciones de trabajo (PC Cliente) utilizan los navegadores *web* Mozilla Firefox e Internet Explorer.

2.4 Arquitectura de la propuesta de solución

En esta sección se describe la arquitectura del buscador implementado, además del API desarrollado para la integración del buscador con la Red Social de la UCI.

2.4.1 Integración del buscador con la Red Social de la UCI

Para lograr la integración del buscador con la Red Social de la UCI, se ha creado un API, la cual brinda las funcionalidades de búsqueda a los usuarios de la red social. El API está estructurado mediante una colección de clases donde cada una de ellas, o en conjunto permiten brindar las funcionalidades que sirven para integrar los servicios del buscador con la Red Social de la UCI, específicamente con el servicio de perfil el cual se describe a continuación:

Servicio de perfil: *plataforma que permite gestionar toda la información asociada a los usuarios, además posibilita la comunicación entre ellos, y la interacción con el resto de los servicios brindados en la Red Social de la UCI.*

2.4.2 Patrón de arquitectura utilizado

"Los patrones de arquitectura son los que definen la estructura de un sistema software, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de este".(Jacobson, Boosh et al. 2000)

A continuación se describe el patrón arquitectónico utilizado.

Arquitectura en capas

"La arquitectura en capas es un estilo arquitectónico de llamada y retorno, en el mismo cada capa proporciona servicios a la capa superior y se sirve de las prestaciones que le brinda la inferior, al dividir un sistema en capas, cada capa puede tratarse de forma independiente, sin tener que conocer los detalles de las demás. La división de un sistema en capas facilita el diseño modular, en la que cada capa contiene un aspecto concreto del sistema y permite además la construcción de sistemas débilmente acoplados, lo que significa que si se minimiza las dependencias entre capas resulta más fácil sustituir la implementación de una capa sin afectar el resto del sistema".(Marquina 2008)

El patrón de arquitectura en 3 capas es básicamente una descripción de los subsistemas, componentes y las relaciones entre ellos. El mismo determina la organización estructural del sistema *software* en tres capas fundamentales (ver **Figura** 8.):

- **Capa de presentación:** es la que interactúa directamente con el usuario, captura la información entrada por éste y hace las peticiones a la capa inferior mostrando al usuario la respuesta proveniente de ésta. Únicamente se comunica con la capa de lógica del negocio.
- **Capa de lógica del negocio:** en esta capa se reciben las peticiones del usuario, y tras ejecutar una acción se le envían las respuestas del proceso. Desde el punto de vista del diseño, esta capa es contenedora de las clases entidades y controladoras. Se comunica con la capa de acceso a datos y brinda información a la capa de presentación, la cual le envía las peticiones y esta le responde con los resultados.
- **Capa de acceso a datos:** es donde se accede a los datos. Se hace referencia a uno o más gestores de BD que realizan el almacenamiento, modificación y consulta de los datos. Recibe peticiones desde la capa de negocios y realiza todas las operaciones de forma transparente para esta otra.

En la capa vista se definen todas las interfaces (páginas HTML) con las que interactúa el usuario para realizar las operaciones del sistema. Cada acción efectuada en la vista es gestionada por un controlador del negocio, el que evalúa el cumplimiento de las precondiciones, gestiona y valida los datos especificados por el usuario y finalmente realiza la operación, consultando el modelo siempre y cuando lo requiera. La vista además incluye los archivos CSS y JS para la definición de reglas, atributos y configuraciones particulares para cada vista. En la capa de negocio, se implementan los servicios del buscador, se definen los controladores y otras clases de utilidad para esta capa. Por otro lado, en la capa de datos, se encuentran aquellas clases que interactúan con el índice de los documentos, además del índice confeccionado después del rastreo realizado.

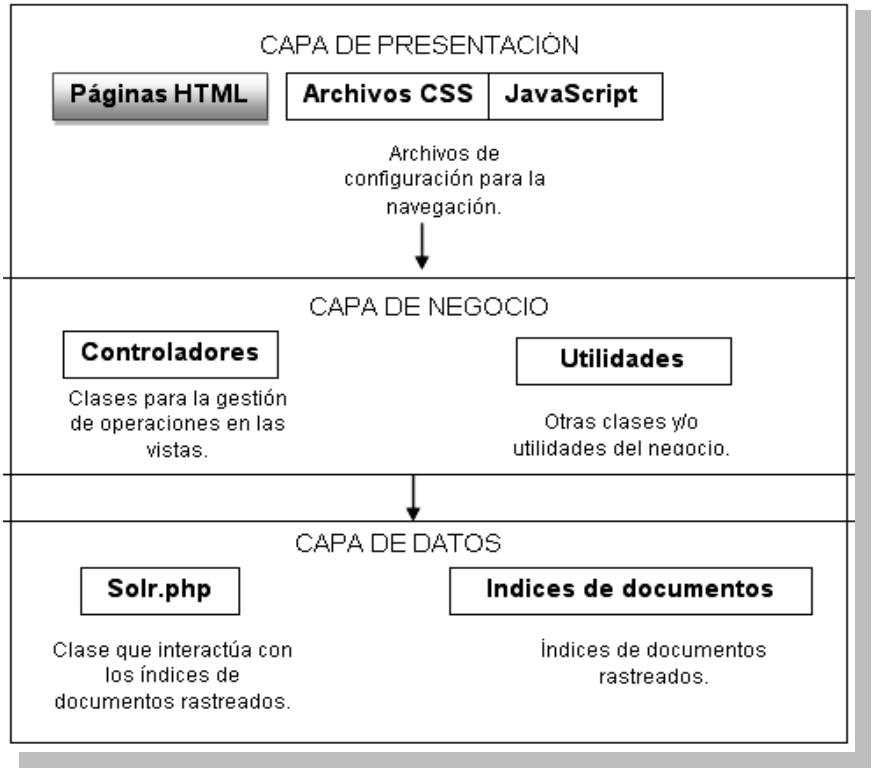


Figura 8. Arquitectura en capas del buscador de la Red Social de la UCI.

2.4.3 Patrones de diseño

En la realización del diseño de aplicaciones en muchos casos se encuentran problemas que resuelven determinados patrones. Los patrones GRASP es un acrónimo que significa *General Responsibility Assignment Software Patterns* (Patrones de *Software* para la asignación General de Responsabilidad). De los diferentes patrones que ofrece GRASP se emplearon para el diseño del buscador de la Red Social de la UCI los siguientes:

Patrón experto: el uso de este patrón garantiza que las clases contengan la información necesaria para cumplir con las responsabilidades por las que fueron creadas, sin depender de ninguna otras, es decir, la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El uso de este patrón se evidencia en la clase Solr, porque ella es la única encargada de todas las funcionalidades relacionadas con la búsqueda de recursos.

Patrón bajo acoplamiento: el acoplamiento viene dado por la fuerza con que una clase está conectada a otras clases**.** Una clase con bajo acoplamiento es cuando dicha clase no depende de muchas otras. En el diseño desarrollado no existe una herencia profunda entre las clases, lo que

significa que una clase no depende de muchas clases, evitando problemas como el difícil entendimiento de ellas y facilitando la reutilización de las mismas, al comunicarse entre ellas lo menos posible.

Alta cohesión: el patrón consiste en asignar una responsabilidad de modo que la cohesión siga siendo alta. En el diseño orientado a objetos, la cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas. Las clases con baja cohesión representan un alto grado de abstracción o han asumido responsabilidades que deberían haber delegado a otros objetos. Este patrón se evidencia en el diseño del sistema porque todas las clases contienen sólo las funcionalidades que están relacionadas con ellas, garantizando que no realicen un trabajo excesivo y que puedan ser fácilmente reutilizables. Un ejemplo de su uso, es en la declaración de una clase para cada servicio del buscador, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, disminuyendo la dependencia entre las clases y por tanto, entre servicios.

2.5 Estándar de codificación

"Los estilos de código constituyen un conjunto de reglas o normas usadas para escribir un programa, que incluyen una gran gama de aspectos dentro del proceso de codificación. Estos permiten que todos los participantes puedan entender el programa en el menor tiempo posible, aportando de esta forma eficiencia en el proceso de desarrollo del mismo y la obtención de programas más robustos y comprensibles".(Desarrolladores 2009) A continuación se describe el estándar de codificación empleado para el desarrollo del sistema.

Identación, llaves de apertura y cierre, y tamaño de las líneas

Usar una identación sin tabulaciones, con un equivalente a 4 espacios, para mantener integridad en las revisiones svn. El uso de las llaves será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75 a 80 caracteres para mantener la legibilidad del código.

```php
/**
 * Retorna el tiempo del sistema
 *
 * @return El tiempo en segundos
 */
function getmicrotime()
{
  list($usec, $sec) = explode(" ",microtime());

  return ((float)$usec + (float)$sec);
}
```

Figura 9. Identación, llaves de apertura y cierre.

Convención de nomenclatura

Variables: se rigen por la nomenclatura *CamelCase*[31]. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

```php
$cantidadResultados = count($result);
$pos = 0;
```

Figura 10. Nomenclatura de variables.

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado "_" y el resto en minúscula.

```php
<?php

class Paginator
{
    var $PaginatorWithNumbers;
    var $PaginatorWithDescription;

 function QuickLink ($linkHref, $desc, $accessKey, $linkTitle)
 {
   $theLink = '<a href="' . $linkHref .'" title="'. $desc .'" accesskey="'.
       $accessKey .'">'. $linkTitle .'</a>';
   return $theLink;

  }
```

Figura 11. Nomenclatura de clases.

Funciones: se rigen por la nomenclatura *CamelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

[31] ***CamelCase*:** estilo de escritura que se le aplica a frases o a palabras compuestas.

```php
/**
 * Retorna el tiempo del sistema
 *
 * @return El tiempo en segundos
 */
function getmicrotime()
{
  list($usec, $sec) = explode(" ",microtime());

  return ((float)$usec + (float)$sec);
}
```

Figura 12. Nomenclatura de funciones.

Estructuras de control

Se incluye *if, for, foreach, while, switch,* entre las estructuras de control. Se utilizan las llaves de apertura y cierre, esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

```php
foreach ($copy_GET as $param => $value)
{
     if($output == "")
     $output = "?".$param."=".quote_replace($value);
     else
     $output .= "&".$param."=".quote_replace($value);
}
```

Figura 13. Estructuras de control.

Documentación

Todos los archivos deben de tener la documentación asociada al mismo. Cada método de igual forma debe tener asociado un comentario donde explique brevemente que es lo que realiza.

Clase:

```
1 /**
2  *Breve descripción de la clase
3  *
4  *PHP versión #
5  *
6  *@category Categoría de la clase implementada “Librería,
7  * Controladora, Modelo”
8  *@package Nombre del paquete o módulo al que pertenece
9  *@author Nombre y Apellidos del autor y correo electrónico.
10*/
```

Figura 14. Documentación de clases.

Funciones:

```
1 /**
2  *Breve descripción de la función
3  *
4  *@param tipo y nombre del parámetro (por cada parámetro
5  * recibe la función)
6  *@return  tipo que retorna
7  *@author Nombre y Apellidos del autor y correo electrónico.
8*/
```

Figura 15. Documentación de métodos.

Buenas prácticas

Los valores booleanos y nulos siempre se escriben con letras mayúscula usando una línea en blanco antes de las estructuras de control y definición de las funciones.

```
1 **** $variableBooleana = FALSE;
2 **** $variableNula = NULL;
3 ****
4 **** if (condición)
5 **** {
6       ****//BI
7 ****}
```

Figura 16. Buenas prácticas.

2.6 Diagrama de despliegue

Los diagramas de despliegue son capaces de describir la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de *software*. La **Figura 17** muestra el diagrama de despliegue, en el que se representan los nodos a utilizar en el despliegue de la aplicación. El sistema está conformado por el nodo PC cliente que se comunicaría mediante el protocolo HTTP con el servidor *web*. Este servidor *web* es considerado como el intermediario entre la PC cliente y el servidor de índices con el que se comunica mediante el protocolo HTTP para gestionar las peticiones realizadas por el usuario.

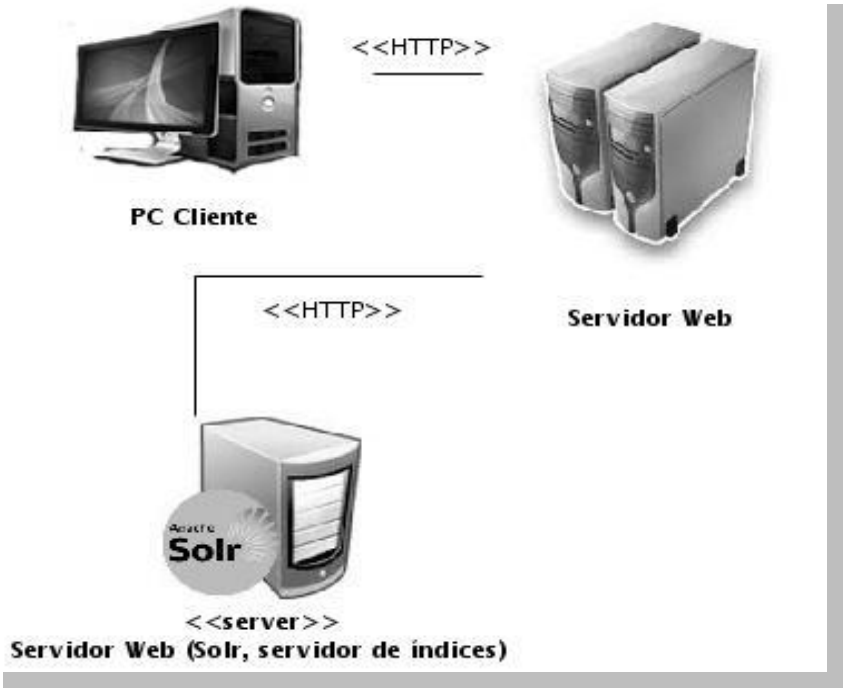


Figura 17. Diagrama de despliegue del buscador de la Red Social de la UCI.

2.6.1 Descripción de los procesadores

PC cliente: ejecuta el buscador con un navegador *web* que puede ser: *Internet Explorer, Mozilla Firefox*, entre otros. Se comunica con el servidor *web* donde se encuentra funcionando el buscador de la RSU de la UCI mediante el protocolo HTTP.

Servidor *web*: es donde se encuentra el buscador funcionando. El servidor de aplicaciones es el que permite que la PC Cliente interactúe y tenga acceso a la aplicación.

Servidor *web* (Solr, servidor de índices): es donde se guardan los índices de los documentos encontrados por el rastreador.

2.7 Conclusiones parciales

Para el desarrollo del buscador de la Red Social de la Universidad de las Ciencias Informáticas se realizó un estudio de la problemática existente. Se definieron las funciones del sistema mediante la descripción de requisitos funcionales y no funcionales, logrando una visión concreta del buscador. Se realizó además un análisis de la arquitectura del sistema, a partir de la descripción de los patrones de diseño y arquitectura, lo que posibilitó un mejor entendimiento del código de la aplicación. Al concluir este capítulo quedan resueltas la mayoría de las tareas propuestas para la elaboración de la solución y el problema planteado.

Capítulo III. Validación del buscador de la Red Social de la Universidad de las Ciencias Informáticas

3.1. Introducción

La evaluación de los SRI no es nueva. Muchos investigadores destacados en el tema han aportado a esta complicada ciencia. Los primeros estudios sobre el tema, se desarrollaron en el Instituto Cranfield de Tecnologías, ubicado en Inglaterra, en el año 1957. Entre los estudios más importantes realizados en Cranfield se encuentran dos, el primero de ellos, dirigido por Cleverdon[32], cuyo objetivo fue comparar la efectividad de cuatro sistemas de indización y el segundo consistió en un experimento controlado destinado a fijar los efectos de los componentes de los lenguajes de indización en la ejecución de los SRI.

"En junio del 2009, se concluyó una exhaustiva investigación realizada por la autora Mireldis García del Valle y en la cual se propone una metodología para la evaluación de los Sistemas de Recuperación de Información Web" (Valle Junio 2009). En la cual, las métricas son agrupadas en tres grandes grupos: las métricas técnicas orientadas a medir de manera concreta y precisa la efectividad del sistema, métricas de calidad para evaluar la calidad de los resultados devueltos así como del proceso de recuperación de información y finalmente métricas orientadas al usuario para poder medir la aceptación del sistema y que tan usable es este para los usuarios. En la evaluación de la solución implementada se utilizaron el conjunto de métricas definidas en dicha investigación.

El presente capítulo tiene como objetivo describir las métricas utilizadas en la evaluación de la solución propuesta. Posteriormente se describen los valores empleados en la ejecución de las pruebas y los resultados obtenidos.

3.2 Métricas técnicas

A continuación se definen las métricas relacionadas con el funcionamiento técnico del buscador propuesto.

[32] **Cyril W. Cleverdon (Bristol, 1914 - Cranfield, 1997),** documentalista científico inglés, pionero de la disciplina Recuperación de información en sistemas documentales. Proporcionó un objeto de estudio, una metodología de investigación y un lenguaje terminológico, siendo el comienzo de investigaciones empíricas y de corte experimental. Además, estableció las directrices apropiadas en la indización automatizada.

3.2.1 Composición de los índices

La composición de los índices afecta de forma muy directa la calidad de la recuperación de información. Es por ello que la composición de los índices siempre ha sido preocupación de la comunidad científica y sobre el cual se han desarrollado distintos criterios, destinados a lograr una mejor optimización de la composición de los índices y, por ende, su consecuente mejora en el rendimiento de los SRI.

En este aspecto se destacan tres componentes importantes: tamaño del índice, frecuencia de actualización y porción de página *web* indizada (título, primeros párrafos, página completa y etiquetas). Las magnitudes de cada motor dependen del *hardware* y el *software* dedicado.

Tamaño del índice del motor de búsqueda (cubrimiento del SRI)

El tamaño del índice o cubrimiento del SRI, es calculado aplicando la siguiente fórmula:

$$Cubrimiento = \frac{PáginasIndizadas}{TotaldePáginas}$$

Figura 18. Fórmula para calcular el tamaño del índice o cubrimiento del SRI.

Las páginas indizadas, son todas aquellas páginas de las cuales existe el índice en la base de datos y por ende, están disponibles en el motor de búsqueda. El total de páginas lo constituyen todas las páginas existentes en la *Web*, cifra que puede resultar inexacta al tener en cuenta el acelerado ritmo de actualizaciones existentes en los sitios *web*, por lo que siempre se toma un valor estimado para el cálculo del tamaño del índice.

Otro elemento a tener en cuenta lo constituye la existencia en la *Web* de sitios cuyo acceso al mismo se encuentra restringido a usuarios registrados, por lo que en muchos casos, su contenido no llega a ser indexado y, por consecuente, nunca están disponibles en el motor de búsqueda.

Frecuencia de actualización

La frecuencia de actualización de los índices del motor de búsqueda, está definida por el tiempo en que expiran los documentos en la base de datos. Cada documento creado en el índice tiene asociado un tiempo de expiración, el cual es definido en la configuración del *spider*. Una vez que los documentos llegan al tiempo de expiración, son marcados como obsoletos, lo que permite que el *spider*, en su recorrido por la *web*, intente obtenerlo nuevamente. Si el *spider* detecta un cambio en el documento, este es actualizado en el índice.

Cada sitio puede tener asociado un tiempo de expiración en la base de datos, lo que permite agrupar los mismos de acuerdo al período en que se actualizan, influyendo positivamente el rendimiento del *spider* en el proceso de descubrimiento de la información y en la calidad en los índices. En el caso de esta investigación, se realiza la clasificación de los sitios atendiendo a la frecuencia de actualización de los mismos, permitiendo actualizar los índices con mucha más frecuencia de aquellos sitios que tienen un ritmo de actualizaciones más acelerado. Tal es el caso de los blogs, portales de proyectos e intranets.

Porción de la página *web* indexada

La porción de la página *web* indexada es definida mediante el sistema de configuración del *spider*. En dicho sistema de configuración, se define un grupo de secciones dentro de la página *web* con sus respectos pesos, teniendo en cuenta la importancia de la sección dentro de la página y las características de la *web* analizada. Dentro de las secciones definidas se pueden encontrar:

- ***Sección title:*** define el título de la página. Es una de las secciones con mayor peso asignado que influye en el orden de relevancia con que se muestran los resultados obtenidos en el motor de búsqueda.
- ***Sección body:*** define el cuerpo de los documentos y es donde mayormente se encuentra la información de la página. El contenido del *body* puede contener textos, imágenes y otros contenidos presentes en la *web*, por lo que es en esta sección, donde radica la verdadera información de valor para el usuario. En esta sección de la página *web* pueden existir contenidos que nunca llegan a ser almacenados en la base de datos, tal es el caso de las animaciones flash, archivos de videos, archivos comprimidos y otros contenidos que no son legibles al proceso de indización.

También se almacenan otras secciones de las páginas *web* tales como: el texto alternativo y las dimensiones de las imágenes, las etiquetas H1 y H2 de HTML y las partes de la dirección del recurso *web*.

3.2.2 Tiempos de respuestas

Los tiempos de respuesta del motor de búsqueda dependen de varios factores. Un factor importante para el buen rendimiento del sistema lo constituye el *hardware* sobre el que se ejecuta. Este tipo de sistema necesita realizar un gran número de consultas a la base de datos, lo que se traduce en un alto consumo del procesador o procesadores que posea el servidor. Otro factor importante, es el tamaño de los índices, a mayor tamaño, mayor número de comparaciones debe hacer el sistema para encontrar y mostrar los resultados esperados por el usuario. La concurrencia también es de vital

importancia. A mayor número de usuarios conectados, mayor cantidad de consultas se procesan en la base de datos, lo que influye en los tiempos de respuestas del servidor.

3.2.3 Especialización en materias

Luego de seleccionar una muestra de los sitios existentes en la *web* de la UCI, dichos sitios fueron clasificados de acuerdo a sus objetivos en distintas categorías, las cuales se relacionan a continuación:

- **producción:** son todos aquellos sitios destinados a la producción, tales como las plataformas Redmine, Alfresco, orientadas a la producción de *software* en la universidad.
- **blogs:** los blogs temáticos constituyen los sitios *web* con más aceptación dentro de la universidad.
- **docencia:** en esta categoría se encuentran las plataformas de tele-formación de la sede principal, las plataformas de las facultades regionales y la plataforma del post-graduado.
- **facultades:** en esta categoría se incluyen los portales de las facultades regionales y la sede principal.

También se incluyeron otras categorías tales como: proyectos, investigación y servicios. Estas categorías o etiquetas, como también se les llama, permiten acotar las búsquedas en los índices a una determinada materia o fin específico.

3.2.4 Prueba piloto

Después de haber realizado las pruebas para evaluar las métricas técnicas, se obtuvieron los siguientes resultados (ver **Tabla 3**), el tamaño del índice es de 90 % lo que indica que se indexaron los sitios existentes en la universidad casi en su totalidad. Otros de los resultados obtenidos fue que le buscador propuesto tiene una frecuencia de actualización entre 3 y 15 días, lo que facilita que la información que muestra como resultado este actualizada. La porción de las páginas *web* se encuentran indexadas a un 95 %, lo que hace que la información existen en el índice tenga una buena estructura. Los tiempos de respuestas del sistema se encuentran entre 0 y 2 segundos, la información del buscador se encuentra agrupada por categorías como son (producción, blogs, docencia, facultades, investigación y servicios), permitiendo acotar las búsquedas en los índices a una determinada materia o fin específico.

Tabla 3. Indicadores evaluados en la prueba piloto.

Indicador	Evaluación
Tamaño del índice	90%
Frecuencia de actualización del índice	3, 7 y 15 días.

Porción de la página *web*	95%
Tiempos de respuestas	entre 0 y 2 segundos
Especialización en materias	agrupamiento por categorías o etiquetas

3.3 Métricas de calidad

A continuación se describen las métricas relacionadas con la calidad del buscador desarrollado.

3.3.1 Calidad de los primeros resultados mostrados

La calidad de los primeros resultados mostrados al usuario, es un indicador medible en el cual se tienen en cuenta dos factores principales:

- **Número de enlaces relevantes:** indica el número de páginas mostradas que realmente se refieren al tema buscado. El número de enlaces relevantes depende en gran medida de la efectividad de los algoritmos empleados por el motor de búsqueda en la asignación de la relevancia a la información contenida en las páginas *web*.
- **Número de enlaces duplicados o muertos:** se consideran enlaces muertos todos aquellos enlaces cuya dirección URL no conducen a ningún lugar de la *web*, es decir, al intentar acceder al recurso *web* en cuestión, se obtiene el código de estado 404 como respuesta del servidor de aplicaciones.

3.3.2 Calidad de los resúmenes

La calidad de los resúmenes de los documentos encontrados depende, en gran medida de la calidad de la información contenida en los mismos. En el caso del motor de búsqueda implementado, para la creación del resumen del documento, se tienen en cuenta algunas secciones importantes del mismo, tal es el caso de la sección *title (título)* y *description (descripción)*. Por lo general, se tienen en cuenta las secciones del documento que contengan mayor número de vocablos de los que forman la consulta del usuario. Estos vocablos encontrados en los documentos, se distinguen del resto del contenido mediante el resaltado de los mismos. A continuación se muestran el comportamiento de las métricas de calidad en las pruebas realizadas.

Indicador\Consulta	**Java**	**Firefox**	**Drupal**	**PHP**
Enlaces relevantes	12	37	28	43
Enlaces muertos	6	0	0	0
Calidad de los resúmenes	Regular	Bien	Bien	Bien

Tabla 4. Comportamiento de las métricas de calidad en la prueba piloto.

Durante el despliegue de una versión piloto del motor de búsqueda se realizaron un grupo de pruebas, para detectar posibles deficiencias en los algoritmos empleados y en las características de la *Web* analizada. En dichas pruebas, se obtuvo 2 de deficiencias que se muestran a continuación:

- Se detectaron la existencia de algunos problemas con el posicionamiento *web* para buscadores, tema conocido como SEO[33].
- Baja actualización de los enlaces existentes en los sitios *web*, lo que provoca que dichos enlaces conduzcan al usuario a sitios no existentes. Estos enlaces también son conocidos como enlaces muertos o enlaces rotos. De un total de 300 sitios encontrados por el *spider*, 99 de ellos son enlaces muertos, lo que representa el 32.1 % del total de enlaces existentes en la *Web* interna de la universidad.

Figura 19. Gráfica que representa el estado de los enlaces en la red interna de la universidad.

3.4 Pruebas de caja negra

Las pruebas de caja negra son aquellas que no tienen en cuenta el código, es decir, el que realiza la prueba no necesita saber cómo está conformado el sistema interiormente o bien no necesita conocer de programación, solo necesita conocer cuáles son las posibles entradas sin necesidad de entender cómo se muestran las salidas. Estas pruebas son realizadas para tratar de encontrar errores en la interfaz.

[33] **SEO:** *Search Engine Optimization* u Optimización para motores de búsqueda.

3.4.1 Casos de pruebas de caja negra

Caso de prueba. Búsqueda básica.

Esta sección cubre el conjunto de pruebas funcionales relacionadas con el requisito de búsqueda básica.

Descripción de la funcionalidad: esta funcionalidad permite la recuperación de aquellos recursos que se encuentran publicados en la red, partiendo de la consulta formulada por el usuario.

Condiciones de ejecución: el servidor de índices tiene que estar funcionando.

Flujo central: escribir la consulta formulada por el usuario, en el campo de búsqueda.

Escenario: "Búsqueda básica".

Tabla 5. Caso de prueba. Búsqueda básica.

Escenario	Descripción	Respuesta del sistema	Flujo central
Búsqueda básica mostrando resultados	El usuario escribe lo que desea buscar en el campo de búsqueda y presiona el botón buscar.	El sistema muestra al usuario las coincidencias encontradas.	El usuario escribe lo que desea buscar en el campo de búsqueda y luego presiona el botón buscar. El sistema busca las coincidencias de la consulta entrada por el usuario en el servidor de índices. El sistema muestra al usuario los resultados y el siguiente mensaje: Resultados **1 - 4** de **4** para **nova** (**0.21** segundos) .
Búsqueda básica sin mostrar resultados	El usuario escribe lo que desea buscar en el campo de búsqueda y presiona el botón buscar.	El sistema muestra al usuario un mensaje de no resultados encontrados	El usuario escribe lo que desea buscar en el campo de búsqueda y luego presiona el botón buscar. El sistema busca las coincidencias de la consulta entrada por el usuario en el servidor de índices. El sistema muestra al usuario el siguiente mensaje: No se encontraron Resultados para **dasdasd** (**0** segundos)
Búsqueda	El usuario	El sistema re-	El usuario deja el campo de búsqueda vacío y

básica con campos vacíos	deja el campo de búsqueda vacío y presiona el botón buscar.	direcciona al usuario hacia la página principal.	presiona botón buscar. El sistema re-direcciona al usuario hacia la página principal.

Caso de prueba. Búsqueda de documentos.

Esta sección cubre el conjunto de pruebas funcionales relacionadas con el requisito de búsqueda de documentos.

Descripción de la funcionalidad: esta funcionalidad permite la recuperación de aquellos documentos que se encuentran publicados en la red, partiendo de la consulta formulada por el usuario.

Condiciones de ejecución: el servidor de índices tiene que estar funcionando.

Flujo central: escribir la consulta formulada por el usuario, en el campo de búsqueda.

Escenario: "Búsqueda de documentos".

Tabla 6. Caso de prueba. Búsqueda de documentos.

Escenario	Descripción	Respuesta del sistema	Flujo central
Búsqueda de documentos mostrando resultados	El usuario selecciona la opción de búsqueda (PDF o Word), luego escribe lo que desea encontrar en el campo de búsqueda y presiona el botón buscar.	El sistema muestra al usuario las coincidencias encontradas.	El usuario selecciona la opción de búsqueda (PDF o Word), luego escribe lo que desea encontrar en el campo de búsqueda y presiona el botón buscar. El sistema busca las coincidencias de la consulta entrada por el usuario en el servidor de índices. El sistema muestra al usuario los resultados y el siguiente mensaje: Resultados **1** - **4** de **4** para **nova** (**0.21** segundos) .
Búsqueda de documentos sin	El usuario selecciona la	El sistema muestra al	El usuario selecciona la opción de búsqueda (PDF o Word), luego escribe lo que desea

mostrar resultados	opción de búsqueda (PDF o Word), luego escribe lo que desea encontrar en el campo de búsqueda y presiona el botón buscar.	usuario un mensaje de no resultados encontrados	encontrar en el campo de búsqueda y presiona el botón buscar. El sistema busca las coincidencias de la consulta entrada por el usuario en el servidor de índices. El sistema muestra al usuario el siguiente mensaje: No se encontraron Resultados para **dasdasd** (**0** segundos)
Búsqueda de documentos con campos vacíos	El usuario deja el campo de búsqueda vacío y presiona el botón buscar.	El sistema re-direcciona al usuario hacia la página principal.	El usuario deja el campo de búsqueda vacío y presiona botón buscar. El sistema re-direcciona al usuario hacia la página principal.

Caso de prueba. Búsqueda de imágenes.

Esta sección cubre el conjunto de pruebas funcionales relacionadas con el requisito de búsqueda de imágenes.

Descripción de la funcionalidad: esta funcionalidad permite la recuperación de aquellas imágenes que se encuentran publicadas en la red, partiendo de la consulta formulada por el usuario.

Condiciones de ejecución: el servidor de índices tiene que estar funcionando.

Flujo central: escribir la consulta formulada por el usuario, en el campo de búsqueda.

Escenario: "Búsqueda de imágenes".

Tabla 7. Caso de prueba. Búsqueda de imágenes.

Escenario	Descripción	Respuesta del sistema	Flujo central
Búsqueda de imágenes	El usuario selecciona la	El sistema muestra al	El usuario selecciona la opción de búsqueda (imágenes), luego escribe lo que desea

mostrando resultados	opción de búsqueda (imágenes), luego escribe lo que desea encontrar en el campo de búsqueda y presiona el botón buscar.	usuario las coincidencias encontradas.	encontrar en el campo de búsqueda y presiona el botón buscar. El sistema busca las coincidencias de la consulta entrada por el usuario en el servidor de índices. El sistema muestra al usuario los resultados y el siguiente mensaje: Resultados **1** - **4** de **4** para **nova** (**0.21** segundos) .
Búsqueda de imágenes sin mostrar resultados	El usuario selecciona la opción de búsqueda (imágenes), luego escribe lo que desea encontrar en el campo de búsqueda y presiona el botón buscar.	El sistema muestra al usuario un mensaje de no resultados encontrados	El usuario selecciona la opción de búsqueda (imágenes), luego escribe lo que desea encontrar en el campo de búsqueda y presiona el botón buscar. El sistema busca las coincidencias de la consulta entrada por el usuario en el servidor de índices. El sistema muestra al usuario el siguiente mensaje: No se encontraron Resultados para **dasdasd** (**0** segundos)
Búsqueda de imágenes con campos vacíos	El usuario deja el campo de búsqueda vacío y presiona el botón buscar.	El sistema re-direcciona al usuario hacia la página principal.	El usuario deja el campo de búsqueda vacío y presiona botón buscar. El sistema re-direcciona al usuario hacia la página principal.

3.4.2 Resultados de las pruebas

El gráfico que se muestra a continuación establece una relación entre las iteraciones de pruebas realizadas y el número de no conformidades detectadas en las mismas.

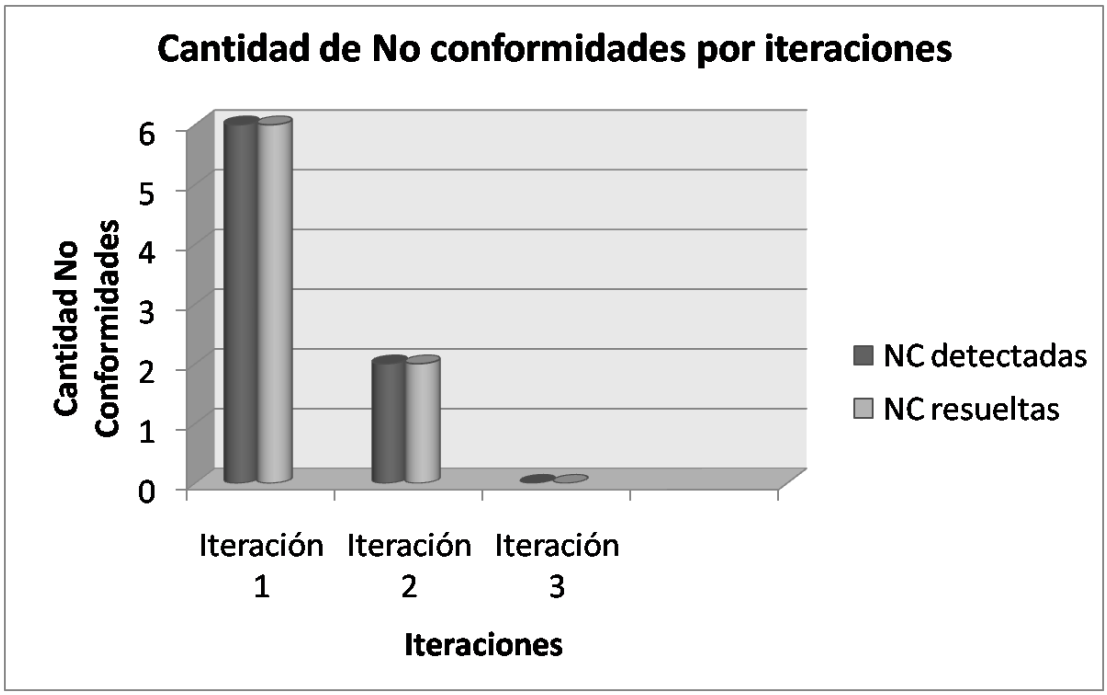


Figura 20. Cantidad de no conformidades encontradas por iteraciones.

En la figura mostrada anteriormente se puede observar que en la primera iteración se detectaron 6 no conformidades, número que se fue reduciendo hasta la tercera iteración donde no se obtuvieron no conformidades. Cada una de las no conformidades detectas fueron resueltas inmediatamente por el equipo de desarrollo. La solución de estas no conformidades ayudó a que la aplicación alcanzara una mejor calidad y cumpliera con los requisitos planteados en el desarrollo del buscador de la Red Social de la UCI.

3.5 Conclusiones parciales

Para la evaluación del sistema de recuperación de información implementado, se empleó la metodología de evaluación propuesta en la tesis de pregrado “Metodología para la evaluación de sistemas de recuperación de información web en la UCI” con la cual se obtuvieron buenos resultados en su aplicación. Se logró la identificación de las fortalezas y debilidades que presenta el sistema desarrollado, además dos deficiencias existentes en la *web* de la universidad. Además se realizaron pruebas sobre la interfaz del software las cuales arrojaron resultados satisfactorios, ya que se obtuvo un número reducido de no conformidades, las cuales fueron solucionadas de forma rápida por el equipo de desarrollo, contribuyendo a obtener un producto de mayor calidad.

Conclusiones generales

La presente investigación tiene un papel fundamental en la definición e implementación del buscador de la Red Social de la Universidad de las Ciencias Informáticas

- Con el estudio de las diferentes funcionalidades de los sistemas de búsqueda y recuperación de la información, se definió el marco teórico conceptual de la investigación, que permitió identificar la situación polémica e identificar las bases para analizar, diseñar e implementar el buscador de la Red Social de la Universidad de las Ciencias Informáticas.
- Con la definición de los requisitos funcionales y no funcionales se logró un mejor entendimiento de las principales funcionalidades del buscador de la Red Social de la Universidad de las Ciencias Informáticas.
- La utilización del sistema implementado contribuye con lo localizar y mostrar a los usuarios de la Red Social de la Universidad de las Ciencias Informáticas todos los recursos publicados en la red.
- Aplicadas las técnicas de validación a la propuesta de solución, se comprobó el buen funcionamiento del sistema de acuerdo a los requisitos planteados.

Al finalizar se obtiene una investigación sustentada en elementos teóricos, todos los artefactos definidos dentro del proceso de desarrollo de software seleccionado y un sistema que permite buscar todos los recursos dispersos y publicados en la red universitaria, por lo que se puede decir que se cumplió con el objetivo principal del trabajo, obteniéndose los resultados esperados.

Recomendaciones

Se recomienda:

- Añadir la funcionalidad que le permita al usuario realizar búsquedas avanzadas sobre el contenido publicado en la Red Social de la Universidad de las Ciencias Informáticas.
- Implementar funcionalidades que permitan al buscador realizar búsquedas semánticas sobre la información publicada en la Red Social de la Universidad de las Ciencias Informáticas.

Bibliografía referenciada

1. Aguirre, J. D. (2007). "Arquitectura de un buscador." Retrieved 16 Febrero, 2012, from http://buscadores.fullblog.com.ar/post/arquitectura_de_un_buscador_531191953898/.
2. Álvarez, M. Á. (2012). Manual de JQuery, http://www.desarrolloweb.com.
3. Baeza-Yates, R. (1999). "Página Web de Ricardo Baeza-Yates." Retrieved 16, Febrero, 2012, from http://www.dcc.uchile.cl/~rbaeza/spanish.html.
4. BAEZA-YATES, R., CASTILLO, C., MARÍN, M. y RODRÍGUEZ, A. (2005). "Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering."
5. Bannister, T. (2012). "What is Apache?". Retrieved 23 de abril, 2012, from http://wiki.apache.org/httpd/FAQ#What_is_Apache.3F.
6. BERGMAN, M. K. (2001). "White Paper: The Deep Web: Surfacing Hidden Value. Ann Arbor." Michigan, Estados Unidos: Universidad de Michigan, Departamento de Publicaciones.
7. Big, m. d. b. (2009). "Big, motor de búsqueda." Retrieved 22 de enero, 2012, from http://es.wikipedia.org/wiki/Bing_%28motor_de_b%C3%BAsqueda%29.
8. BOJO, C. (2004). "Internet Visible e Invisible: búsqueda y selección de recursos de información en Ciencias de la Salud." Instituto de Salud Carlos III, Madrid.
9. bvs.sld.cu (2007). "Buscador cubano en internet." Retrieved 20 de noviembre, 2011, from http://bvs.sld.cu/revistas/aci/vol15_05_07/aci16507.htm.
10. 10.CASTILLO, C. (2004). Effective Web Crawling. Facultad de Ciencias Físicas y Matemáticas, Departamento de Ciencias de la Computación. Santiago, Chile:, Universidad de Chile**:** 194.
11. CERN (2008). "Where the web was born." Retrieved 10 Febrero, 2012, from http://public.web.cern.ch/public/en/About/Web-en.html.
12. Clavijo, I. K. R. (2011). "Concepción del sistema Red Social Académica de la UCI " Centro de Informatización Universitaria: 15.
13. Códova, C. (2007). Requisitos funcionales. Cumaná, Centro Local Sucre.
14. Códova, C. (2007). Requisitos no funcionales. Cumaná, Centro Local Sucre.
15. Comin, J. (2001). "La historia de Yahoo." Retrieved 16 Febrero, 2012, from http://www.maestrosdelweb.com/editorial/yahoohis/.
16. Consoft (2007). "¿Qué son los metabuscadores?". Retrieved 16 Febrero, 2012, from http://www.consoft.es/noticias/news_text.asp?id=33219.
17. Desarrolladores (2009). "¿Qué es un estándar de codificación?". Retrieved 22 de enero, 2012, from http://mx.answers.yahoo.com/question/index?qid=20090122200540AAOI1N8.

18. Estrada, A. S. M. Y. S. M. A. D. (2012). Procedimiento para el desarrollo de software con un enfoque ágil y CMMI nivel 2. La Habana, Universidad de las Ciencias Informáticas.
19. Group, P. (2009). "PHP: Hypertext Preprocessor." Retrieved 16 Febrero, 2012, from http://php.net/.
20. GUTIÉRREZ, C. (2008). Cómo funciona La Web. Santiago, Chile, Universidad de Chile, Centro de Investigación de la Web.
21. Herrera, D. P. (2007). "IDE."
22. Jacobson, I., G. Boosh, et al. (2000). El proceso Unificado de Desarrollo de Software. Imprenta Fareso S.A, Addison Wesley Logan Inc.
23. Kiva (2009). "Buscadores o Search Engine." Retrieved 16 Febrero, 2012, from http://www.searchoptimization.es/buscadores-search-engines/buscadores-search-engines.htm.
24. Lazo, M. J. R. and M. S. Reyes (2007). Módulo Control de Acceso del proyecto Intranet del Centro Rector de Universidad para Todos La Habana, Universidad de las Ciencias Informáticas.
25. López Herrera, A. G. (2006). Modelos de Sistemas de Recuperación de Información Lingüística Difusa.
26. Marquina, E. (2008). Arquitectura en capas. Estados Unidos, Microsoft Corporation.
27. Masadelante.com. "HTML." Retrieved 18, Diciembre, 2011, from http://www.masadelante.com/faqs/html.
28. Netbeans (2012). "NetBeans IDE 7.0.1 Release Information." Retrieved 20 de enero, 2012, from http://netbeans.org/community/releases/70/.
29. Nioche, J. (2009). "Features - Nutch Wiki." Retrieved 16 Febrero, 2012, from http://wiki.apache.org/nutch/Features.
30. Nutch, T. (2009). "About Nutch." Retrieved 16 Febrero, 2012, from http://lucene.apache.org/nutch/about.html#Overview.
31. Paradigm, V. (2012). "Visual Paradigm for UML." Retrieved 20 de noviembre, 2011, from http://www.visualparadigm.com.
32. Pérez, J. E. (2008). Introducción a CSS, Creative Commons Reconocimiento - No Comercial - Sin Obra Derivada 3.0.
33. Pérez, J. E. (2008). Introducción a JavaScript, Creative Commons Reconocimiento - No Comercial - Sin Obra Derivada 3.0.

34. PÉREZ, Y. y. R., D. A (2008). Asignación automatizada de categorías temáticas al contenido textual de documentos HTML. Facultad 10. Ciudad de La Habana, Universidad de las Ciencias Informáticas**:** 94.
35. Pinto, M. (2004). "Búsqueda y Recuperación de Información." Retrieved 16 Febrero, 2012, from http://www.mariapinto.es/e-coms/recu_infor.htm#ri1.
36. Sooft, I. (2007). Requerimientos de Software.
37. Stoughton, N. (2005). "Update on Standards."
38. Swish, T. (2010). "Swish-e :: Home Page." Retrieved 16 Febrero, 2012, from http://swish-e.org/index.html.
39. Valle, M. G. d. (Junio 2009). METODOLOGÍA PARA LA EVALUACIÓN DE SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN WEB EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS. Facultad 10. Ciudad Habana, Universidad de las Ciencias Informáticas**:** 108
40. www.posicionamiento-web.org (2009). "Historia de Altavista." Retrieved 19 de enero, 2012, from http://www.posicionamiento-web.org/news/historia_de_altavista/2009-09-05-21.
41. www.taringa.net (2009). "La diferencia entre Google y Yahoo.". Retrieved 20 de enero, 2012, from http://www.taringa.net/posts/info/2027308/La-diferencia-entre-Google-y-Yahoo!.html

Bibliografía consultada

1. ARASU, A.; CHO, J., et al. Searching the *Web*. 2001, nº Disponible en: http://oak.cs.ucla.edu/~cho/papers/cho-toit01.pdf.
2. CENTRO DE INVESTIGACIÓN LA *WEB*, U. D. C. Cómo funciona la *Web*. Editado por: Gallardo, E. G. C. G. Primera Edición, Junio 2008 ed. Santiago de Chile: © 2008 Centro de Investigación la *Web*, 2008, ISBN ISBN: 978-956-319-225-1.
3. DELGADO, I. Y. H. Herramientas FOSS para buscadores empresariales. 1 diciembre 2011, nº p. 30. [Consultado el: 10 diciembre del 2011].
4. DELGADO, Y. H. Orión, un motor de búsquedas para la *web* de la UCI. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2010.
5. ESTRADA, A. S. M. Y. S. M. A. D. Procedimiento para el desarrollo de software con un enfoque ágil y CMMI nivel 2. La Habana: Universidad de las Ciencias Informáticas, 2012, Disponible en: http://uciencia.uci.cu/es/node/1310.
6. GALLARDO, A. W. M. y ESTRADA, S. S. Recolector de contenido *web*. Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2009.
7. GÓMEZ, A. H. Directorio del Buscador CubaSearch. Trabajo para optar por el título de Ingeniería en Informática, Unviersidad de las Ciencias Informáticas, 2006.
8. HEYDON, A. y NAJORK, M. Mercator: A Scalable, Extensible *Web* Crawler. nº [Consultado el: 28 de noviembre del 2011]. Disponible en: http://cgi.di.uoa.gr/~ad/MDE519.docs/scalable-crawler.pdf.
9. LUNA, J. F. y GUADIX, J. H. Empleo de motores de búsqueda de código abierto para la recuperación de información vertical. Revista Cubana de Ciencias Informáticas, 2009, vol. 3, nº 1-2, p. 41-47. [Consultado el: 12 diciembre del 2011].
10. MANNING, C. D. An Introduction to Information Retrieval. Cambridge UP, 2007, nº [Consultado el: 20 de enero del 2012].
11. MARTÍNEZ COMECHE, J. A. Los modelos clásicos de recuperación de información y su vigencia. 2008.
12. MARTÍNEZ MÉNDEZ, F. J. Propuesta y desarrollo de un modelo para la evaluación de la recuperación deinformación en internet. 2002.
13. NUTCH. OJOBuscador [Consultado el: 12 de enero de 2012]. Disponible en: http://www.ojobuscador.com.

14. PINTO, M. Búsqueda y Recuperación de Información [Consultado el: 16 Febrero de 2012]. Disponible en: http://www.mariapinto.es/e-coms/recu_infor.htm#ri1.
15. PULIDO, S. M. Modelo de Recuperación Probabilístico [Consultado el: 16 Febrero de 2012]. Disponible en: http://modelosderecuperacioni.iespana.es/probabilistico.html.
16. SALAZAR, C. D. B. y ROMERO, E. M. Sis*web* Sistema Bot-*Web* Buscador e Indexador de Información. Universidad de las Ciencias Informáticas, 2008.
17. SUÁREZ MOLINA, J. Spider UCI. Tesis de Ingenieria, Universidad de las Ciencias Informáticas, 2004.

Glosario de términos

Código abierto: es una tendencia internacional del desarrollo de *software* basada en la distribución del código junto a las aplicaciones.

Contenido *web*: todo documento, imagen, animación, sonido, video, que pueda ser transmitido y ejecutado a través de un navegador.

Enlace: elemento de una página *web* que hace referencia a otro documento.

Internet: red de ordenadores a nivel mundial que ofrece distintos servicios, como el envío y recepción de correo electrónico, la posibilidad de ver información en las páginas *web*, de participar en foros de discusión, de enviar y recibir ficheros, de charlar en tiempo real, entre otros.

Metodología: se refiere a los métodos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia. Son un conjunto de métodos que se rigen en una investigación científica o en una exposición doctrinal.

Ranking: tabla o lista en que se clasifican una serie de elementos por orden de mayor a menor categoría o puntuación: Ejemplo: esta película encabeza el ranking de las películas más premiadas de la historia del cine; el número uno del ranking mundial de tenis ha sido eliminado por un jugador que ocupa el número 90 de la clasificación.

Servidor: en informática, un servidor es un tipo de *software* que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al computador en el cual funciona ese *software*, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

URL: secuencia de caracteres de acuerdo a un formato estándar, que se usa para nombrar recursos como documentos o imágenes en Internet, según su localización.

Web: término utilizado para referirse a Internet, o WWW. Se utiliza *web* para referirse a las páginas *web* que conforman la Internet o intranet.

Anexos

Como instalar Nutch en Ubuntu.

Prerrequisitos: Es preciso tener previamente instalado Java SDK 1.5 ó superior. Para ello, se puede consultar el manual "Como instalar JAVA JDK en Ubuntu

".

Proceso de Instalación:

Paso 1: se copia el archivo en su lugar de instalación en este caso se hará en: /opt/.

Paso 2: se descomprime el archivo creando el directorio Nutch dentro de la dirección donde se va a instalar:

/opt/nutch/

Paso 3: verificar la correcta instalación de Nutch. Para ello, ir a /opt/nutch/ runtime/local tecleando en un terminal:

$ cd /opt/nutch/runtime/local

Una vez allí, permitiremos la ejecución del programa al usuario que habitualmente empleará el programa. Considerando que es el mismo que ha seguido los pasos hasta aquí, basta teclear en un terminal:

$ chmod +x bin/nutch

De igual forma, debemos comprobar que la variable de entorno JAVA_HOME está correctamente configurada. Para ello, consultar el tutorial "Cómo instalar Java SDK en Ubuntu 9.10", en especial los pasos 15-20.

Efectuadas estas comprobaciones, ejecutamos el programa Nutch tecleando:

$ bin/nutch

La instalación es correcta si se observan las siguientes líneas:

Usage: nutch [-core] COMMAND

where COMMAND is one of:

…….

NOTE: this works only for jobs executed in 'local' mode

Paso 4: Configurar inicialmente el programa para poder realizar rastreos de la web. En primer lugar, se incluye el nombre del agente -buscador, el equipo de investigación, la Facultad, etc.- que llevará a cabo el crawling. Esta información sobre quién lleva a cabo las tareas de rastreo debe añadirse a uno

de los archivos de configuración del programa. Suponiendo que el agente es "buscador de la RSU de la UCI", primeramente se situará en:

$ cd /opt/nutch/runtime/local/conf

A continuación, para conservar la versión inicial del archivo nutch-site.xml, se renombrará:

$ mv nutch-site.xml nutch-site-old.xml

Luego se hace una copia del archivo nutch-default.xml para emplearlo como archivo nutch-site.xml, que a su vez se modificará posteriormente para configurar el programa adaptándolo a las necesidades de sus usuarios:

$ cp nutch-default.xml nutch-site.xml

Ahora se edita el archivo nutch-site.xml:

$ gedit nutch-site.xml

Y se modifica la propiedad "agent.name" de manera que quede:

<property>

<name>http.agent.name**</name>**

<value>Buscador de la RSU de la UCI**</value>**

<property>

Se guardan los cambios efectuados en el archivo.

Paso 6: Crear un archivo de texto plano con la/las url/urls inicial/iniciales que se emplearán a modo de semillas (seeds) para rastrear la web. Por ejemplo, para rastrear el sitio web de la intranet de la universidad, se crea un archivo denominado "lista" con la siguiente línea:

http://intranet2.uci.cu

Para ello, se crea un directorio denominado "urls" directamente bajo nutch/runtime/local:

$ mkdir /opt/nutch/runtime/local/urls

Se crea el archivo "lista":

$ gedit /opt/nutch/runtime/local/urls /lista

Una vez que se haya tecleado la línea "http://intranet2.uci.cu/", se cierra el archivo (Ctrl-S).

Paso 7: en estos momentos ya están la condiciones para efectuar un primer rastreo del sitio web de La intranet de la universidad. Para ello se situará en:

$ cd /opt/nutch/runtime/local

$ bin/nutch crawl urls -dir crawl -depth 3 -topN 5

Se habrá creado el directorio nutch/runtime/local/crawl con los siguientes subdirectorios: crawldb, linkdb y segments, donde se hallan los resultados del rastreo del sitio web de la intranet de la universidad.

Paso 8: Para comprobar cuántas urls hay en la base de datos y cuántas han sido rastreadas, se ejecuta el siguiente comando en un terminal:

```
$ bin/nutch readdb /opt/nutch/runtime/local/crawl/crawldb -stats
```

Se mostrará en pantalla una información semejante a esta:

```
TOTAL urls: 413
….....
status 1 (db_unfetched): 402
status 2 (db_fetched): 11
CrawlDb statistics: done
```

Como instalar Solr en Ubuntu.

Prerrequisitos: Es preciso tener previamente instalado Java SDK 1.5 ó superior. Para ello, se puede consultar el manual "Cómo instalar Java SDK en Ubuntu".

Paso 1: copiar el archivo al lugar donde vaya a instalarse, en nuestro caso, /opt/. Para ello tecleamos en un terminal:

Paso 2: Descomprimir el archivo en el lugar donde se vaya a instalar Solr. Se creará el directorio /opt/solr/ **Paso 3:** Solr puede funcionar con cualquier contenedor de servlets, como Tomcat, pero es suficiente con emplear Jetty, cuya instalación es muy simple. Para ello teclear en un terminal:

```
$ cd /opt/solr/example
$ java -jar start.jar
```

Se mostrará un mensaje parecido a este en pantalla:

```
08-jul-2011 19:26:47 org.apache.solr.core.SolrCore registerSearcher
INFO: [ ] Registered new searcher Searcher@18efaea main
….....
2011-07-08 19:26:47.488:INFO:Started SocketConnector@0.0.0.0:8983
```

No se debe dar a ninguna tecla, aunque el cursor esté parpadeando. De hecho, con este último comando se ha iniciado Jetty en el puerto 8983, y está en funcionamiento. Se limitará, pues, a minimizar el terminal, de manera que siga funcionando Jetty, y se abre el navegador introduciendo la dirección:

```
http://localhost:8983/solr/admin/
```

Si el proceso se ha efectuado correctamente, se observará la pantalla inicial de la consola de administración del programa Solr (que se ha iniciado también junto con Jetty). De igual forma, si se introduce la dirección:

http://localhost:8983/solr/admin/stats.jsp

Se tendrá una pantalla con toda la información acerca del programa Solr y de la colección cargada (la primera vez, lógicamente, indica numDocs: 0, esto es, que no hay ningún documento cargado en el sistema). Para cerrar la pantalla y consiguientemente el programa Solr, basta teclear Ctrl-C en el terminal que se tiene minimizado.

Paso 4: Una vez que tanto Nutch como Solr se han instalado y configurado correctamente, se deben de integrarlos de manera que las URLs obtenidas con Nutch puedan ser recuperadas mediante Solr. Para ello, se teclea en un terminal:

$ cp /opt/nutch/runtime/local/conf/schema.xml /opt/solr/example/solr/conf/

de manera que se reemplace el archivo schema.xml por defecto de Solr con el de Nutch.

Paso 5: Reiniciar Solr (junto con la consola web facilitada por Jetty) con el comando empleado en el **Paso 3:**

$ cd /opt/solr/example

$ java -jar start.jar

Se minimiza el terminal para seguir teniendo acceso a la consola web.

Paso 6: Abrir otro terminal, dejando el anterior minimizado, e introducir el comando de Nutch que efectúa la indexación de las URLs rastreadas en el Paso 7 anterior. Para ello, teclear en el nuevo terminal:

$ cd /opt/nutch/runtime/local

$ bin/nutch solrindex http://127.0.0.1:8983/solr/ crawl/crawldb crawl/linkdb crawl/segments/*

en caso de que esta forma no funciones se procede a realizarlo de la siguiente manera, se teclea en un terminal:

$ cd /opt/nutch/runtime/local

$ bin/nutch crawl urls/lista -solr http://127.0.0.1:8983/solr/ -depth 5 –topN 10

Este comando consigue que Solr indexe todos los datos del rastreo efectuado en el Paso 7. En pantalla se observará un mensaje semejante a este:

SolrIndexer: starting at 2011-07-08 20:13:24

SolrIndexer: finished at 2011-07-08 20:13:26 elapsed: 00:00:02

Paso 7: Si el proceso se ha efectuado correctamente, se puede empezar a realizar búsquedas sobre esas páginas. Para ello, se abre el navegador y se introduce la dirección:

http://localhost:8983/solr/admin/

En Query String se puede introducir, por ejemplo, la búsqueda del término “nutch”:

Query String: +nutch

Se hace clic en el botón “Search” y se obtiene un fichero en XML con los documentos (páginas web rastreadas) que satisfacen esa consulta.

Como instalar JAVA JDK en Ubuntu

Paso 1: Comprobar que está instalado el paquete “java-package”, bien con aptget o con el programa Synaptic. Con este último, por ejemplo, se elige Sistema → Administración → Gestor de paquetes Synaptic. En la ventana que surge, seleccionar Editar → Buscar. En el área de texto, introducir “java-package” y pinchar en “Buscar”. Si está instalado, aparecerá con el cuadrado a la izquierda del paquete en verde. Si no está instalado, aparecerá con el cuadrado a la izquierda del paquete en blanco. Para instalarlo, emplear “aptget install” o el propio Synaptic.

Con este último, por ejemplo, se marca con el ratón el cuadrado a la izquierda del paquete “java-package”, y en el menú “Paquete” se selecciona “Marcar para instalar”. Si deben instalarse paquetes adicionales aparecerán en la nueva ventana que surge. En ese caso, pinchar en “Marcar”. A continuación, hacer clic en “Aplicar”. En la ventana de confirmación, pinchar en “Aplicar”.

Paso 2: Por defecto, se instalará donde suele hacerse normalmente, esto es, en /usr/lib/jvm. Para ello, en primer lugar, se debe comprobar que el directorio /usr/lib tiene todos los permisos otorgados al usuario habitual del sistema, en este caso “yoennis”, y no a “root”. Para comprobarlo, en un terminal se teclea:

```
# cd /usr
# ls l
```

Tanto el propietario como el grupo del directorio “lib/jvm” debe ser “yoennis” y no otro como “root”. Es decir, debe aparecer una línea como la siguiente:

```
drwxrxrx
202 yoennis yoennis 69632 20091108
19:18 lib
```

Si en lugar de “yoennis” aparece “root”, se tendrá que modificar. Para ello, en un terminal se teclea:

```
# sudo chown R
yoennis:yoennis /usr/lib/jvm
```

Paso 3: Comprobar, mediante el gestor de paquetes Synaptic, que el programa java-6-openjdk está perfectamente instalado. Para ello, en Sistema → Administración → Gestor de paquetes Synaptic, seleccionamos Editar → Buscar, e introducimos java-6-openjdk Tras pinchar en “Buscar” deberá

aparecer dicho paquete con el cuadrado a la izquierda en verde y a la derecha con la versión recién instalada

Paso 4: Establecer el Java SDK como nuestra Máquina Virtual por defecto. Para ello instalamos previamente el paquete "galternatives":

```
# sudo aptitude install galternatives
```

Contestar "Y" cuando pregunte. Se recibirá el siguiente mensaje final: "Escribiendo información de estado extendido... Hecho". También se puede instalar con Synaptic, siguiendo un proceso similar al señalado en el Paso 1.

Paso 5: Lanzar el programa galternatives:

```
# sudo galternatives
```

Paso 6: En el menú gráfico que se obtiene, en el marco izquierdo se selecciona "java" y en el marco derecho se selecciona la nueva versión de Java si aparecen varias, esto es, /usr/lib/jvm/java-6-openjdk/ bin/java. A continuación, cerrar el programa (Fichero → Salir).

Paso 7: Comprobar que se está usando la versión correcta, esto es, la recién instalada. Para ello, basta teclear en un terminal:

```
# sudo java version
```

y se obtiene:

```
java version "1.6.0_17"
Java ™ SE Runtime Environment (build.....)
Java HotSpot ™ Server VM (build ……)
```

Paso 8: se procede a configurar las variables JAVA_HOME y CLASSPATH. Para el establecimiento de estas variables de entorno, basta modificar el archivo ".bashrc", localizado y oculto en el directorio raíz de cada usuario. Por ejemplo: /home/yoennis/.bashrc, donde yoennis" es el nombre de la carpeta de usuario. Abrir con gedit el archivo ".bashrc" tecleando en un terminal:

```
sudo gedit /home/yoennis/.bashrc
```

Paso 19: Al final del archivo se añaden las siguientes líneas de texto:

```
JAVA_HOME='/usr/lib/java-6-openjdk/
CLASSPATH='/usr/lib/java-6-openjdk/lib:/home/yoennis:.'
export CLASSPATH
export JAVA_HOME
PATH=$JAVA_HOME/bin:$PATH
export PATH
```

La tercera variable de entorno PATH tiene una finalidad importante. De hecho, para poder ejecutar programas Java desde cualquier carpeta, la carpeta con los archivos “javac.exe” y “java.exe” debe estar incluida en el PATH del sistema.

Paso 20: Una vez añadidas estas líneas al final del archivo, se guardan los cambios. De esta manera, se ha instalado y configurado la Máquina Virtual Java adecuadamente.

Paso 21: Es importante reiniciar el sistema para que todas las variables de entorno adquieran los valores recién introducidos. Para comprobar que todo está correcto, se abre de nuevo un terminal y se teclea “echo $JAVA_HOME”. Deberá mostrar la ruta introducida anteriormente. De igual forma, si se teclea “echo $CLASSPATH” deberá mostrar las rutas tecleadas previamente.