Universidad de las Ciencias Informáticas

Facultad 15

Orión, un motor de búsquedas para la web de la UCI

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

Autor

Yusniel Hidalgo Delgado

Tutores

Ing. Abdiel Matos Nieto

Ing. Eduardo Manuel Macias Sotolongo

Ciudad de la Habana, Junio del 2010

"Año 52 de la Revolución"

Declaración de autoría

Yo, Yusniel Hidalgo Delgado, declaro que soy el único autor de este trabajo y autorizo a la Facultad 15 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmo el presente a los _____ días del mes de _____ del año _____.

Autor	Tutor	Tutor
Yusniel Hidalgo Delgado	Ing. Abdiel Matos Nieto	Ing. Eduardo M. Macias Sotolongo
_____	_____	_____
Firma del autor	Firma del tutor	Firma del tutor

Ing. Abdiel Matos Nieto

Graduado de Ingeniero en Ciencias Informáticas por la Universidad de las Ciencias Informáticas en el año 2009. Actualmente se encuentra laborando en la Universidad de las Ciencias Informáticas como profesor con la categoría docente de Instructor Recién Graduado.

Correo electrónico: anieto@uci.cu

Ing. Eduardo Manuel Macias Sotolongo

Graduado en el 2008 de Ingeniería en Ciencias Informáticas en la UCI. Adiestrado, Jefe del Grupo de Desarrollo del Proyecto GIDI. Profesor del Departamento de Técnicas de Programación de la facultad 10, 2 años de experiencia.

Correo electrónico: emmacias@uci.cu

Agradecimientos

Primeramente quiero agradecer a la Revolución y a mi Comandante en Jefe Fidel Castro, agradezco el haberme formado y hacer de mi un hombre de bien.

Agradezco a mi familia toda, sin ellos, este sueño no hubiese sido posible. Los amo mucho.

Agradezco a mis tutores Abdiel y Eduardo. Gracias por haberme apoyado cuando tomé una decisión importante en mi vida.

Agradezco a mis mejores profesores, a todos los que me enseñaron todo el conocimiento que hoy poseo. En especial a Arián Cabezas Regal y Odette Fernández, ya es prácticamente como mi hermana.

A mis colegas de causa, Abel Meneses, Eduardo Estevez. ¡Viva el Software Libre!

A mis amigos todos, siempre aprendo de ellos. En especial a Mary, Roxy, Grety, Ilianita, Dayneris, Adrian y Yoel.

A mis compañeros y amigos Liván y Yasmany.

A todos los que me apoyaron durante la realización de esta tesis.

A mi Comandante en Jefe Fidel Castro Ruz.

A mi madre que, aunque lejos, siempre la guardo muy dentro de mi corazón. Gracias por hacer de mi la persona que soy hoy.

A papi, sin él, cumplir este sueño no hubiera sido posible. Te quiero mucho.

A mis hermanos Mita y Papo por confiar en mí, por apoyarme cuando más lo he necesitado.

A mis sobrinos Manuel y Alejandra, los quiero mucho.

A mi tía Tata, la persona más dulce y cariñosa que he conocido. Te quiero mucho, mucho. Gracias por escucharme, prometo escucharte siempre.

A mi familia toda.

Resumen

Muchos expertos coinciden en que la información existente en la empresa u organización, es considerada como el activo intangible más importante de la misma, incluso, más importante que los activos tangibles. Es por ello que una adecuada gestión de la misma dentro de la organización, en ocasiones, equivale al logro de metas mucho más ambiciosas.

El presente trabajo de diploma tiene como objetivo desarrollar un Sistema de Recuperación de Información, específicamente un motor de búsqueda, que permita optimizar la búsqueda y recuperación de la información existente en la red interna de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo planteado en la investigación, se realizó el estado del arte referente a los principales motores de búsqueda existentes, todos con licencias de software libre, fundamentalmente la licencia GPL. Se investigó además, acerca de las principales herramientas y tecnologías que más se adecuan a la solución deseada.

Se realizó el diseño y la implementación de la solución propuesta obteniéndose un producto de software con los resultados y calidad esperados dando cumplimiento así a los objetivos planteados inicialmente. Para la evaluación del sistema obtenido, se empleó una metodología de evaluación de sistema de recuperación de información propuesta en una tesis de pregrado desarrollada en nuestra propia universidad.

Como parte del proceso de evaluación y prueba piloto realizada al sistema, se detectaron dos deficiencias fundamentales presentes en la web de la universidad, la primera relacionada con el bajo posicionamiento web y optimización para buscadores y la segunda relacionada con el bajo nivel de actualizaciones de los enlaces existentes en los sitios web.

Índice de contenido

Introducción

Es en los primeros años de la década de los 90 del pasado siglo, cuando Tim Berners-Lee acuña el término World Wide Web, en lo adelante WWW. Es entonces cuando la WWW evoluciona hasta convertirse en el primero de los servicios que ofrece la red de redes. En esa época, se produce también la aparición de la Internet comercial, seguido, las empresas hacen su aparición en Internet ofreciendo todo tipo de servicios en línea: tiendas, bancos, etc.

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.

Los sistemas de recuperación de información, en lo adelante SRI, constituyen el mecanismo ideal para resolver este tipo de problemas. Estos permiten localizar y procesar la información de forma rápida y en forma automática. Son sistemas capaces de localizar cualquier contenido existente en la web, tales como textos, imágenes, videos, archivos de sonido, entre otros. En este sentido destacan los directorios temáticos, los motores de búsqueda o buscadores y los meta buscadores.

Desde el surgimiento mismo de la Internet, la vida de muchos cambió para siempre. Se vive en lo que se le ha dado en llamar, la sociedad de la información y el conocimiento [1]. El desarrollo económico, tecnológico y social de una nación, está cada vez más ligado al desarrollo del conocimiento científico y tecnológico de la misma. Cuba no está exenta de esta regla.

En Cuba, se está llevando 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, 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, entidad con significativos aportes a la informatización del país.

Como parte de los programas de formación e informatización de la sociedad y al calor de la batalla de ideas que libra nuestro pueblo, surge en el año 2002 la Universidad de las Ciencias Informáticas, en lo adelante UCI. Dicha universidad posee una infraestructura tecnológica bastante avanzada respecto a otras tecnologías existentes en el país.

En la UCI, actualmente existe una red LAN con más de 10000 usuarios, los cuales tienen acceso a más de 300 sitios web con más de 1 millón de documentos. Cada año se puede evidenciar un notable incremento en los contenidos de la red, los cuales son confirmados en los estudios webmétricos realizados por el Polo de Soluciones Informáticas para Internet de la facultad 10.

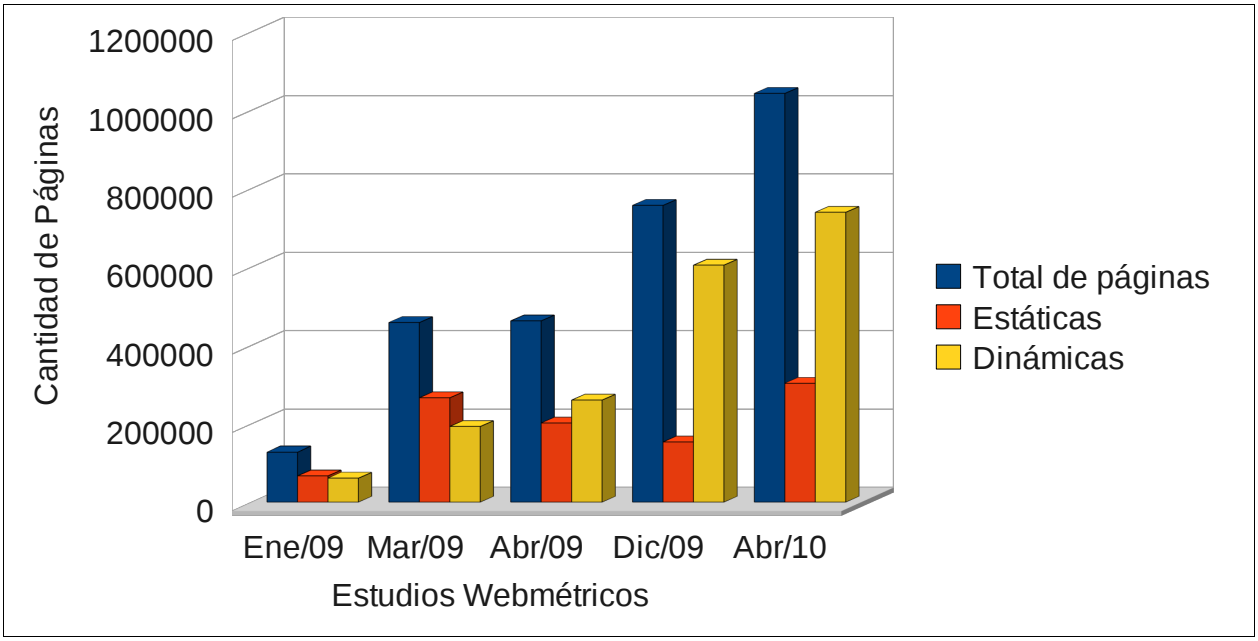

Figura 1. Resultados obtenidos en los estudios webmétricos realizados en la UCI.

Aunque muchos de estos sitios poseen un motor de búsqueda interno, las opciones de búsqueda en la red LAN se ven limitadas, lo cual provoca que se pierda tiempo y recursos buscando la información requerida. Ante esta disyuntiva, muchos usuarios se ven necesitados de hacer uso de la cuota de Internet asignada por los servicios telemáticos de la Universidad para acceder a información, que muchas veces ya se encuentra en la propia red interna. Todo esto trae consigo un mal uso del canal de Internet del que se dispone.

Teniendo en cuenta la **situación problemática** descrita anteriormente se enuncia el siguiente **problema a resolver**: ¿Cómo optimizar la búsqueda y recuperación de la información existente en la red LAN de la UCI que posibilite un mejor aprovechamiento del canal de Internet del que se dispone?.

El **objeto de estudio** de la investigación lo constituyen los Sistemas de Búsqueda y Recuperación de la Información y el **campo de acción** se define como: los motores de búsqueda.

Para darle solución al problema descrito, se ha planteado el siguiente **objetivo general**:
Implementar un motor de búsqueda capaz de optimizar la búsqueda y recuperación de la información existente en la UCI.

Del cual se desglosan los siguientes **objetivos específicos**:

1. Realizar el estado del arte sobre los sistemas de búsqueda y recuperación de la información en la web.
2. Diseñar un motor de búsqueda que cumpla con los requisitos solicitados con las tecnologías propuestas.
3. Codificar la solución informática previamente diseñada.
4. Comprobar la correcta ejecución del motor de búsqueda.

Para dar cumplimento a los objetivos específicos planteado con anterioridad, se definen las siguientes

tareas de investigación:

1. Estudio de la bibliografía existente sobre los principales sistemas de recuperación de la información existentes para definir cuál o cuáles de ellos utilizar en la solución a implementar.
2. Estudio de los principales algoritmos utilizados para establecer un orden de relevancia de la información indizada.
3. Identificación de los principales requerimientos para la implementación del motor de búsqueda.
4. Definición de la base tecnológica a utilizar en la implementación del motor de búsqueda que satisfaga los principales requerimientos identificados.
5. Descripción de los casos de usos a implementar por el motor de búsqueda.
6. Implementación del motor de búsqueda.
7. Realización de pruebas al motor de búsqueda para verificar su correcto funcionamiento.

Idea a defender:

Con la implementación de un motor de búsqueda en la UCI se optimizará la búsqueda y recuperación de información en la web de la Universidad.

Resultados esperados:

Se pretende que al concluir la investigación, la red interna de la Universidad de las Ciencias Informáticas cuente con motor de búsqueda que satisfaga las necesidades básicas de información de los usuarios partiendo de la información disponible en los sitios web existentes en la red.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En este capítulo se exponen algunos criterios valorativos sobre los sistemas de recuperación de información existentes, su teoría e importancia en el desarrollo informacional de las instituciones. Se expondrán además, algunos de los motores de búsqueda de código abierto que más se utilizan, sus características, ventajas y desventajas, así como los algoritmos fundamentales que implementan y que permiten obtener un mejor desempeño del sistema. Se mostrarán algunos de los avances obtenidos en este sentido tanto a nivel internacional, nacional y en la UCI.

1.2 Los sistemas de recuperación de información

Muchos son los autores que, a lo largo de estos años, han intentado dar una definición exacta de qué es recuperación de información. Aunque todos los autores observan el término desde distintas aristas, todos concluyen en esencia en que la recuperación de información difiere sustancialmente de la recuperación de datos. En este trabajo, se describirá el concepto dado por Ricardo Baeza-Yates[2] el cual sí hace una distinción más concisa entre recuperación de la información o *information retrieval* y recuperación de datos o *data retrieval*.

Para Baeza-Yates *"los datos se pueden estructurar en tablas, árboles, para recuperar exactamente lo que se quiere, el texto no posee una estructura clara y no resulta fácil crearla" [3]* .

Por otra parte, este autor considera la recuperación de la información como *"dada una necesidad de información (consulta) y un conjunto de documentos, ordenar los documentos de más a menos relevantes para esa necesidad y presentar un conjunto de aquellos con mayor relevancia" [3]* .

En la solución de este problema se identifican dos grandes etapas [4]:

1. Elección de un modelo que permita calcular la relevancia de un documento frente a una consulta.
2. Diseño de algoritmos y estructuras de datos que implementen este modelo de forma eficiente.

1.3 Arquitectura de un SRI

Un sistema de recuperación de información se define como el *"p*roceso donde se accede a una información previamente almacenada, mediante herramientas informáticas que permiten establecer ecuaciones de búsquedas específicas. Dicha información ha debido ser estructurada previamente a su almacenamiento*"[5]*.

Generalmente, todos los sistemas de recuperación de información comparten una misma arquitectura, la cual se detalla a continuación [6]:

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.

Sistema de Formulación de Consultas: realiza un preprocesamiento 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 preprocesada para obtener un subconjunto de documentos relevantes que satisfagan la consulta introducida por el usuario, ordenados estos de acuerdo a un criterio de relevancia.

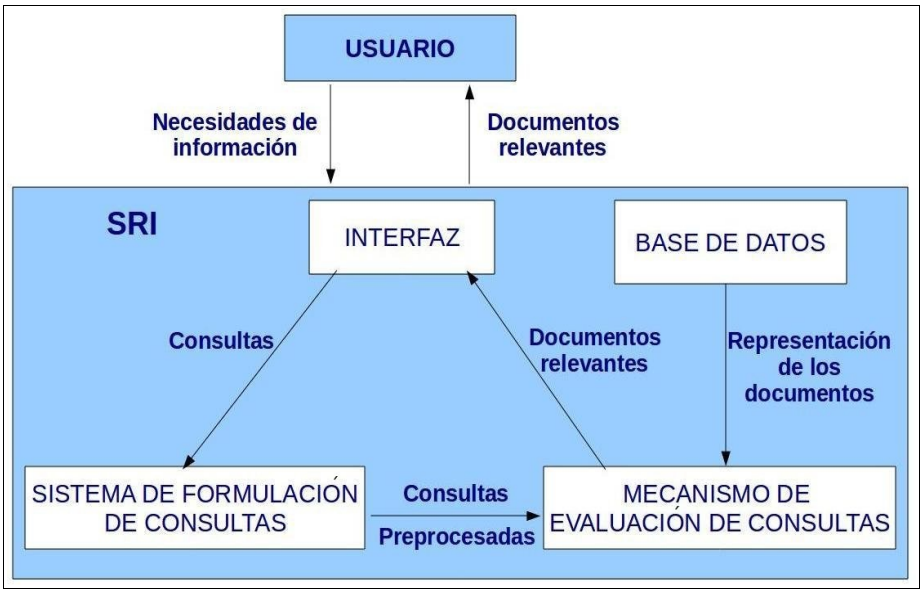

Figura 2. Arquitectura de un Sistema de Recuperación de Información

1.4 Sistemas de Recuperación de Información existentes

Algunos de los principales sistemas de recuperación de información existentes en Internet son: los buscadores, los directorios y los metabuscadores, en los cuales se profundizará en la siguiente sección.

1.4.1 Buscadores

Los buscadores, son sistemas de recuperación de la información que permiten, dado un criterio de búsqueda introducido por el usuario, obtener un subconjunto de aquellos documentos de mayor relevancia para dicho criterio de búsqueda, mediante la realización de ciertas operaciones sobre una base de datos.

Estos sistemas consideran 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.

Algunos de estos obstáculos son [7]:

- La información existente en la web, no sigue una estructura bien definida lo que implica que la información se encuentre desordenada.
- La información cambia continuamente.
- La información es redundante.

Arquitectura general de un buscador

La mayoría de los buscadores emplean una arquitectura araña-indizador centralizada, es decir, la araña y el componente de indización se encuentran unidos [4].

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 tales como: HTTP, FTP, NNTP 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[8]

Muchos indizadores emplean técnicas avanzadas para la extracción de vocabulario tales como[8]:

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

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 y 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 originalmente introducida por el usuario[6].

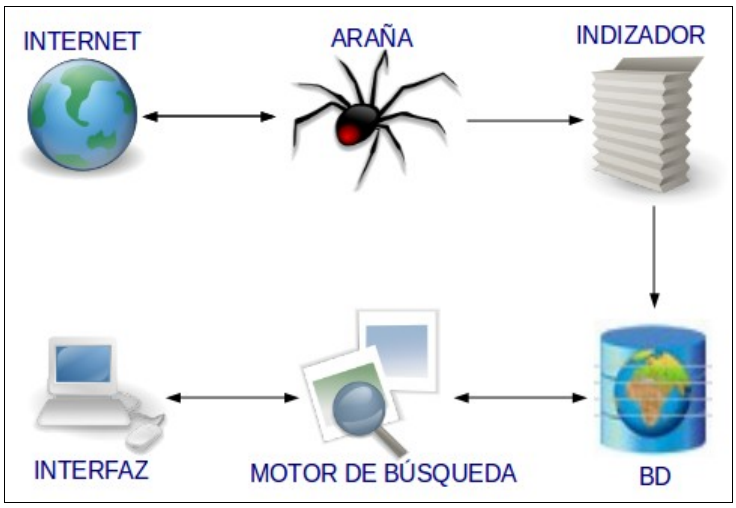

Figura 3. Arquitectura de un buscador.

1.4.2 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 una 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 necesitamos. 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[9].

1.4.3 Metabuscadores

A diferencia de los buscadores, un metabuscador no posee una base de datos propia sino que utiliza la base de datos de estos para encontrar la información solicitada por el usuario. 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[10].

Hay que tener en cuenta que 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 porque coincidir en los del resto. Aportando puntos de vista distintos[10].

1.4.4 Free For All

Free For All, en lo adelante FFA, no son más que páginas donde los usuarios envían su dirección URL con un título. El alta es inmediata, apareciendo la URL enviada en el primer elemento de la lista de las URL temporales. El nuevo enlace va descendiendo de posición hasta que finalmente desaparece. En algunos FFA con abundante tráfico, las primeras URL, suelen desaparecer en un término de 24 horas. Al principio, estos sistemas fueron muy útiles, pero han perdido importancia con la utilización de altas automatizadas mediante programas de envío a buscadores [11].

1.4.5 Buscadores verticales

Un buscador vertical es un buscador especializado en un sector o nicho concreto, lo que le permite analizar la información con mayor profundidad que un buscador genérico, disponer de resultados más actualizados y ofrecer al usuario herramientas de búsqueda avanzadas[12].

Para otorgar cierta relevancia a los documentos contenidos en las bases de datos, estos sistemas de recuperación de información utilizan modelos matemáticos, los cuales serán tratados en la siguiente sección[12].

1.5 Modelos matemáticos de los SRI

Existen varios modelos matemáticos o técnicas empleadas en la recuperación de información, y, como en todo, cada uno de ellos tiene sus ventajas e inconvenientes. En esta sección se abordan dichos modelos analizando las componentes que los forman. Los principales modelos clásicos de recuperación de información son: modelo Booleano, modelo Espacio Vectorial, modelo Probabilístico y modelo Booleano extendido o modelo Difuso.

1.5.1 Modelo Booleano

Es un modelo de recuperación simple, basado en la teoría de conjuntos y el álgebra booleana. Dada su inherente simplicidad y su pulcro formalismo ha recibido gran atención, siendo adoptado por muchos de los primeros sistemas bibliográficos comerciales. Su estrategia de recuperación está basada en un criterio de decisión binario (pertinente o no pertinente) sin ninguna noción de escala de medida, sin noción de un emparejamiento parcial en las condiciones de la pregunta[13].

En un sistema de este tipo, los documentos se encuentran representados por conjuntos de palabras claves (términos). La indización se realiza asociando un peso binario a cada término del índice: 0 si el término no aparece en el documento y 1 si aparece al menos una sola vez. Las búsquedas consisten en un conjunto de palabras claves conectadas por los operadores lógicos **AND, OR, NOT**. El grado de similitud entre un documento y una consulta también será binario y un documento será relevante cuando su grado de similitud sea igual a 1, de lo contrario, el documento no tendrá ninguna relevancia en cuanto a la consulta[6].

De este modelo inicial han surgido variaciones como el “booleano extendido”, que asigna pesos a los términos de búsqueda, o el “fuzzy”, que ha incluido la lógica difusa entre sus postulados [14].

1.5.2 Modelo Probabilístico

El modelo probabilístico de recuperación de información fue propuesto en 1976 por los autores Robertson y Sparck-Jones. Este modelo, también conocido como *Binary Independence Retrieval* intenta resolver el problema de la recuperación de información desde una óptica probabilística. Se basa fundamentalmente en estimar la probabilidad de que un documento sea relevante para una consulta dada, o sea, dada una consulta, existe un conjunto de documentos que contienen exactamente los documentos relevantes y no otros[14][15].

Matemáticamente, la similitud entre un documento y una consulta es[16]:

$$sim(d, c)=P(R|d) / P(R'|d)$$

donde:

P(R|d) es la probabilidad de que el documento *d* sea relevante en el conjunto de documentos relevantes R y ***P(R'| d)*** es la probabilidad de que el documento *d* no sea relevante en el conjunto de documentos relevantes R, por lo que:

Un documento será relevante si ***P(R|d) > P(R'|d)***.

La probabilidad de que un documento sea relevante para una consulta dada, depende de la forma en que se representan los documentos en el sistema. Una vez calculada la probabilidad de los documentos relevantes para la consulta dada, son ordenados descendentemente y mostrados al usuario[16].

1.5.3 Modelo Espacio Vectorial

El modelo de Espacio Vectorial, propuesto por Gerald Salton a finales de los años 60, es uno de los modelos que me se emplean en los sistemas de recuperación de información en la actualidad. En este modelo, cada documento se representa como un vector t-dimensional, donde t corresponde a la cantidad de términos asociados al documento. Cada elemento del vector, tomará un valor de relevancia (peso) dependiendo de cuán representativo o no sea el término en el documento. Un término que no aparezca en el documento, tendrá un peso igual a 0[17].

Existen varias formas para calcular el peso asociado a un término dentro de un documento, siendo la más extendida la ponderación **TF.IDF** que consiste en multiplicar dos factores que reflejan la importancia de los términos en el documento:
El primer factor **TF** (abreviatura de ***Term Frequency***), pretende reflejar la importancia de los términos en los documentos, concediendo mayor importancia a los términos que aparecen con mayor frecuencia en los documentos[16].

El segundo factor **IDF** (abreviatura de ***Inverse Document Frequency***), o Inverso de la Frecuencia de Documentos, dará mayor importancia a un término cuanto menor sea el número de documentos de la colección en los que aparece dicho término. Es decir, el IDF de un término es inversamente proporcional al número de documentos en que aparece dicho término[16].

Si un término aparece mucho en un documento su peso aumentará, pero si aparece en muchos documentos, disminuirá, pues este término no es muy útil para distinguir un documento de otro[18].
Así mismo, se crea un vector para cada una de las consultas introducidas por los usuarios con las mismas características de los vectores por cada documento de la colección. Esto permite calcular, fácilmente un valor de similitud entre el vector de una consulta y los vectores de cada uno de los documentos. La similitud entre dos documentos será dada por la distancia coseno[18]:

$$sim(d_i, d_j) = \vec{d_i} \cdot \vec{d_j} = \sum_{r=1}^{k} w_{r,i} \times w_{r,j}$$

Así mismo, se puede calcular la relevancia de los documentos para una consulta de acuerdo a la siguiente fórmula:

$$R(\vec{q}, \vec{d_i}) = \cos(\widehat{\vec{q}, \vec{d_i}}) = \frac{\langle \vec{q}, \vec{d_i} \rangle}{|q| \cdot |d_i|} = \frac{\sum_{j=1}^{n} w_{j,q} \cdot w_{j,i}}{\sum_{j=1}^{n} w_{j,q}^2 \cdot \sum_{j=1}^{n} w_{j,i}^2}$$

Lo cual permite establecer un orden en que se muestran los documentos recuperados por el sistema de recuperación de información.

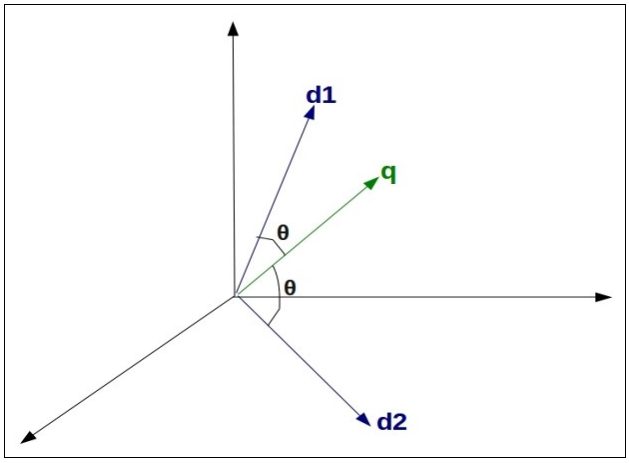

Figura 4. Representación matemática del modelo de espacio vectorial.

1.6 Algunos buscadores existentes

Actualmente existen un grupo de herramientas que permiten realizar distintos tipos de búsqueda de

manera efectiva. En esta sección se dará a conocer algunos detalles correspondientes a las herramientas de búsqueda más utilizadas tanto nacionales, internacionales y dentro de la UCI.

1.6.1 Buscadores Internacionales

En el ámbito internacional destaca el buscador **Google**. Google es el producto más notorio de una empresa norteamericana del mismo nombre. Fue fundado en 1998 por Larry Page y Serguei Brin, dos estudiantes de doctorado de la Universidad de Stanford. Desde entonces, la empresa a logrado situarse entre las punteras en cuanto a desarrollo tecnológico se refiere, a tal punto, que es considerado un monopolio tecnológico y por ende, ha sido objeto de sanciones financieras durante algunos años[19].

Por otro lado se tiene a su competidor **Yahoo**, el cual fue fundado en 1994 por Jerry Yang y David Filo, ambos estudiantes graduados de la Universidad de Stanford. Yahoo comenzó siendo una lista de sitios realizada por los autores. A medida que navegaban por la web, recopilaban las direcciones de los sitios y los clasificaban a mano, sin la intervención de un sistema automático. Poco a poco fue adquiriendo notoriedad entre los internautas de la época, hasta convertirse en lo que hoy se conoce como Yahoo, una empresa que ha tenido muy buena aceptación con sus productos y servicios[20].

También existen otros buscadores bastante interesantes. Tal es el caso de WolframAlfa[1], un buscador un tanto atípico, pues permite realizar consultas y obtener resultados con cierto grado de inteligencia, como puede ser, evaluación de funciones matemáticas y respuestas concretas a preguntas que implican cierta lógica interna.

1 http://www.wolframalpha.com/index.html

1.6.2 Buscadores nacionales

En el caso específico de nuestro país contamos con varios directorios temáticos, tal es el caso del directorio turístico[2]. Este directorio turístico contiene un grupo de informaciones y sitios de interés para nuestros visitantes extranjeros, tales como: hoteles, playas, centros culturales y recreativos, entre otros.

Otro proyecto de buscador interesante lo constituye el buscador cubano 2x3[3], un directorio temático y motor de búsqueda orientado a la red de sitios nacionales, fundamentalmente, los sitios de la prensa cubana. Fue desarrollado por especialistas de la ONI[4], entidad perteneciente el Ministerio de la Informática y las Comunicaciones de Cuba y lanzado en el marco de la Feria y Convención Internacional Informática 2007.

1.6.3 Buscadores en la UCI

Faro fue desarrollado por un grupo de estudiantes y profesores de la facultad 10 de la UCI en el año 2008. Para la recolección de las páginas web, utilizaba un spider llamado Wire de origen chileno y para la creación de los índices, un indizador 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. Gugle es una parodia del motor de búsqueda más utilizado en Internet. Surgió en el año 2008 como respuesta a una creciente necesidad de buscar y encontrar información útil en los sitios existentes en nuestra universidad. Actualmente el proyecto está descontinuado.

2 http://www.dtcuba.com/
3 http://www.2x3.cu/
4 ONI: Ofincina Nacional para la Informatización

1.7 Metodologías, herramientas y técnicas utilizadas

1.7.1 Htdig

Ht://Dig 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[21]. El proyecto se descontinuó en junio del año 2004. Fue desarrollado en C/C++. Su curva de aprendizaje es alta.

Entre sus principales características destacan:

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

1.7.2 Swish-e

Swish-e es un sistema rápido, flexible y de código abierto para la indexación de colecciones de páginas web y otros archivos[22].
Entre sus principales características destacan:

- Soporte para archivos de texto plano, HTML, PDF, e-mail y XML.
- Soporte para el SGBD MySQL.
- Fichero de índice portable entre plataformas.

- Soporte para búsqueda de frases.
- Desarrollado en C con algunos script en Perl.
- Posee un script CGI para realizar búsquedas sobre los índices generados.

La última versión estable de Swish-e es la 2.4.7 y fue lanzada el 5 de abril del 2009. Su licencia es GPL y posee versiones tanto para GNU/Linux como para Windows. Posee una amplia comunidad aunque la documentación en idioma español es escasa[22].

1.7.3 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. También permite, haciendo uso de analizadores de varios formatos de archivos, extraer información desde archivos PDF, XML, HTML, y otros[23].
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.
Entre sus principales características destacan[24]:

- Captura, parser e indización en modo paralelo y distribuido.
- Extensible mediante plugins.
- Soporte para diversos formatos, tales como: texto plano, HTML, XML, ZIP, ODF, PDF; JS, RSS, etc.
- Ontologías.
- Solución basada en cluster.
- Sistema de fichero distribuido.
- Soporte para autenticación NTLM.

1.7.4 MnoGoSearch

mnoGoSearch es un motor de búsqueda completo de código abierto y basado en SQL. mnoGosearch consiste en dos partes. La primera parte es un mecanismo de indización -indexer- el cual se mueve a través de vínculos de hypertexto HTML y almacena información acerca de los documentos en la base de datos. La segunda parte es una interfaz web CGI search.cgi la cual muestra en el navegador un formulario HTML y los resultados de búsquedas. search.cgi utiliza información recopilada por el indizador[25].
Entre sus principales características destacan:

- Soporte para diversos protocolos: HTTP, HTTPS, FTP, NNTP.
- Analizadores incorporados para diversos formatos de archivo: text/html, text/xml, text/plain y audion/mpeg.
- Soporte para autenticación de proxy.
- Indización multihilo.
- Interfaces web CGI, Perl y PHP.
- Lenguaje de consulta booleano.
- Soporte para la mayoría de los conjuntos de caracteres modernos.
- Soporte para múltiples bases de datos: MySql, PostgreSQl, SQLite, Mimer, Virtuoso, Interbase, Oracle, MS SQL, DB2, Sysbase, etc.
- Posee una API externa para PHP.
- Manejo de clústeres de base de datos.
- Fácil de configurar.

mnoGoSearch es software libre, lanzado bajo la licencia GPL de la **Free Software Fundation**. Integra la mayoría de los conceptos y características asociadas a los motores de búsqueda actuales. Para el cálculo de la relevancia de los documentos, emplea los algoritmos clásicos Espacio Vectorial y Booleano. Ambos

algoritmos han sido ampliamente probados y utilizados, obteniéndose muy buenos resultados con los mismos[25].

Luego de un exhaustivo análisis sobre los spiders investigados, se optó por la utilización del spider contenido en mnoGoSearch para la implementación del motor de búsqueda, teniendo en cuenta las características y facilidades que posee.

1.7.5 Metodología de desarrollo

Como metodología de desarrollo de software, se utiliza el Proceso Unificado de Rational (RUP) y como lenguaje de modelado el UML, el cual es propuesto por dicha metodología. Ambos son ampliamente utilizado en el desarrollo de software orientado a objeto a nivel mundial. RUP se caracteriza por dividir el ciclo de vida del desarrollo del software en 4 fases y 9 flujos de trabajo, 6 de los cuales son destinados a la ingeniería y 3 al soporte [26].

1.7.6 Lenguajes de programación

PHP

PHP es un lenguaje de programación interpretado de propósito general. Es ampliamente utilizado en el desarrollo web y puede ser embebido en páginas HTML. La última versión estable de PHP es la versión 5.2.3 [27].

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.

Características

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas [28]:

- Soporte para una gran cantidad de bases de datos: MysSQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF, analizar código XML, entre otras.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que se puede realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Java

Java es un lenguaje de programación orientado a objeto desarrollado por Sun Microsystem a principio de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria [29].

Hasta la fecha, la plataforma Java ha atraído a más de 6.5 millones de desarrolladores alrededor del mundo y está presente en un elevado número de dispositivos, equipos y redes. Se caracteriza por su portabilidad y seguridad, lo cual la han convertido en la tecnología ideal para su aplicación a redes [29].

1.7.7 Frameworks de desarrollo

Symfony

Symfony es un completo framework para el desarrollo ágil de complejas aplicaciones web. Está completamente desarrollado en PHP 5. Symfony está basado en la experiencia, no reinventa la rueda, sino que usa las mejores prácticas del desarrollo web e integra una gran variedad de librerías[30].
Posee una activa comunidad de usuarios y desarrolladores, los cuales desarrollan plugins que extienden las funcionalidades del framework. Existe abundante documentación en diferentes idiomas, fundamentalmente en inglés, español y francés.

Zend Framework

Zend Frameworks está basado en la simplicidad y las mejores prácticas de la Programación Orientada a Objetos. Enfocado en la construcción de aplicaciones web seguras, fiables y orientadas a la web 2.0, consume un considerable grupo de APIs que incluyen integración con Google, Amazon, Yahoo y otras. Está siendo desarrollado por Zend Technologies Ltd [31].

Kumbia

KumbiaPHP es un framework para aplicaciones web libre escrito en PHP5. Basado en las prácticas de desarrollo web como DRY y el Principio KISS para software comercial y educativo. Kumbiaphp fomenta la velocidad y eficiencia en la creación y mantenimiento de aplicaciones web, reemplazando tareas de codificación repetitivas por poder, control y placer [32].
KumbiaPHP Framework intenta proporcionar facilidades para construir aplicaciones robustas para

entornos comerciales. Esto significa que es muy flexible y configurable.
KumbiaPHP es un esfuerzo por producir un framework que ayude a reducir el tiempo de desarrollo de una aplicación web sin producir efectos sobre los programadores[32].

Sus principales características son [32]:

- Sistema de Plantillas sencillo
- Administración de Caché
- Scaffolding Avanzado
- Modelo de Objetos y Separación MVC
- Soporte para AJAX
- Generación de Formularios
- Componentes Gráficos
- Seguridad

Adicional a esto, Kumbia integra lo mejor de la Web en un solo framework para producir las aplicaciones Web (prototypejs, phpMailer, Smarty, FPDF, Script.aculo.us).

1.7.8 Abstracción de la Base de Datos

Symfony posee varios plugins que permiten integrar completamente en el framework dos de los principales ORMs existentes para PHP. Dichos ORMs se describen a continuación:

Doctrine

Doctrine es un ORM[5] para PHP 5.2.3 o superior situado sobre una potente capa de abstracción de base de datos (DBAL). Una de sus principales características consiste en la posibilidad de escribir consultas de base de datos en un lenguaje SQL orientado a objetos llamado Doctrine Query Language (DQL) inspirado en HQL de Hibernate. Doctrine está construido principalmente utilizando los patrones: Active Record, Data

5 ORM: Object Relational Mapper o Mapeo Objeto Relacional.

Mapper y Meta Data Mapping[33].
La versión estable del ORM al momento de escribir este artículo es la versión 1.2.2.

Propel

Propel es un ORM de código abierto desarrollado para PHP 5. Permite el acceso a diferentes sistemas de gestión de base de datos utilizando un conjunto de objetos que proveen al desarrollador de una API[6] simple para la obtención y almacenamiento de datos en una base de datos relacional. Propel utiliza la capa de abstracción de base de datos PDO[7], lo cual lo convierte en un ORM muy rápido. Implementa las mejores prácticas en la programación de ORM, haciendo uso de los patrones más comunes en este tipo de software: ActiveRecord, validators, behaviors, table inheritance, etc [34].

1.7.9 Sistemas de Gestión de Base de Datos

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales[35].

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando[35].

MySQL

MySQL es el Sistema de Gestión de Base de Datos de código abierto más popular del mundo, con más de 100 millones de copias a lo largo de su historia. Con su velocidad, fiabilidad y facilidad de uso, se ha

6 API: Application Programming Interface - Interfaz de Programación de Aplicaciones.
7 PDO: PHP Data Objects - Capa de abstracción de base de datos para PHP.

convertido en la elección predilecta por los desarrolladores web. MySQL es parte importante de LAMP (Linux, Apache, MySQL y PHP), toda una suite o compendio de aplicaciones de código abierto con amplia aceptación por las empresas [36].

1.8 Conclusiones parciales

Teniendo en cuenta las características presentes en cada una de las herramientas descritas con anterioridad, se define la siguiente base tecnológica a utilizar en el desarrollo del motor de búsqueda.

Como ya se había comentado, mnoGoSearch posee una API para el popular lenguaje de programación PHP. Esta API brinda un conjunto de funciones en PHP que permiten interactuar con la base de datos que contiene los índices generados en el proceso de indización de la información obtenida por el *spider*. Teniendo en cuenta este aspecto, se define el lenguaje de programación PHP para el desarrollo del motor de búsqueda Orión.

Como framework utilizado en el desarrollo del sistema, se define el framework Symfony, el cual se destaca por poseer una arquitectura flexible, la cual es extensible mediante plugins, característica esta que hereda el motor de búsqueda en aras de enriquecer su desarrollo con la inclusión de nuevas y atractivas funcionalidades. Por su parte, se determinó el uso de PostgreSQL como Sistema Gestor de Base de Datos.

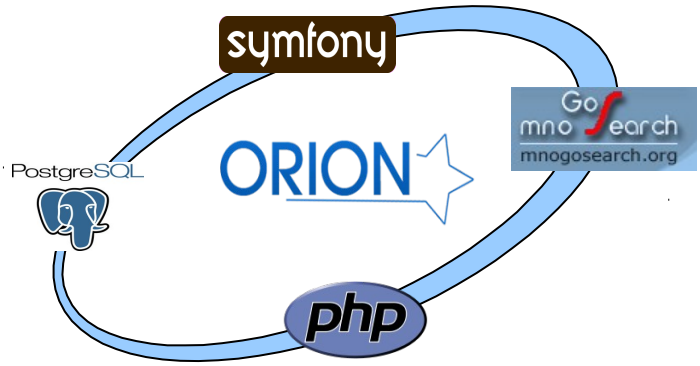

Figura 5. Base tecnológica empleada en el desarrollo del proyecto.

Capítulo 2. Diseño e implementación del sistema

2.1 Introducción

En este capítulo se definen los requerimientos de software, las estructuras de datos y los artefactos necesarios para la implementación de la solución propuesta. Se detalla el sistema desde el punto de vista ingenieril mediante el modelo de diseño realizado.

2.2 Propuesta de sistema

Se propone el desarrollo del sistema basado en una arquitectura Cliente-Servidor. La arquitectura Cliente-Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura [37].

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes[37]:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas de la arquitectura cliente-servidor[37]:

- El servidor no necesita potencia de procesamiento, parte del proceso se reparte con los clientes.

– Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

Se propone un sistema con una estructura modular, permitiendo que los distintos módulos sean implementados por separado para posteriormente integrarlos en un todo. Esta estructura en forma modular, posee algunas ventajas tales como:

– Fácil integración con otros sistemas de aplicaciones favoreciendo la reutilización del código.
– Flexibilidad en las actualizaciones de los módulos. Una actualización crítica en un módulo, no debería tener profundos cambios en el funcionamiento del sistema como un todo.

2.3 Requerimientos del sistema

El sistema, una vez implementado, deberá responder a un conjunto de requisitos definidos previamente y los cuales se detallan a continuación.

2.3.1 Requisitos funcionales

– Búsqueda básica.
  La búsqueda básica debe permitir la recuperación de aquellos recursos que se encuentran en la web, partiendo de la consulta formulada por el usuario.
– Búsqueda de imágenes y documentos.
  Búsqueda por tipo de contenido. Debe recuperar todos los archivos con las extensiones de imágenes y documentos.
– Obtener noticias.
  El sistema debe ser capaz de obtener las noticias de los canales RSS disponibles en la Universidad.
– Mostrar las noticias obtenidas.

Mostrar al usuario, los datos básicos de las noticias obtenidas mediante los canales RSS.

– Calificar las noticias.

  Las noticias mostradas al usuario, deben ser evaluadas positiva o negativamente por el usuario, lo cual influye en la relevancia de la noticia y, por ende, en la posición dentro de las noticias destacadas del sistema.

– Generar boletín de noticias en formato PDF.

  Dado un rango de fecha, debe permitir la generación de un boletín de noticias en formato PDF, conteniendo las noticias introducidas al sistema en el rango de fecha definido.

– Gestionar perfil de usuario.

  El usuario debe tener la posibilidad de gestionar su perfil de usuario.

– Adicionar bookmarks al perfil de usuario.

  Permite adicionar al perfil un resultado obtenidos mediante el motor de búsqueda.

– Editar bookmarks del perfil de usuario.

  Permite editar los bookmark de los resultados obtenido mediante el motor de búsqueda.

– Eliminar bookmarks del perfil de usuario.

  Permite eliminar del perfil, un bookmark previamente añadido.

– Autenticar usuario en el sistema.

  El sistema debe brindar la posibilidad de que los usuarios puedan autenticarse en el sistema.

– Enviar resultado de búsqueda por correo.

  El sistema de permitir el envío de resultados encontrados mediante el motor de búsqueda a un destinatario mediante el correo electrónico.

2.3.2 Requisitos no funcionales

Usabilidad

El sistema debe permitir acceder a sus distintas partes con una profundidad máxima de 3 clics.

Disponibilidad

El sistema debe permanecer disponible las 24 horas del día. En caso de la ocurrencia de una falla técnica o eventualidad no prevista, se cuenta con un máximo de 2 horas para la corrección del problema y la puesta en línea del sistema nuevamente.

Rendimiento

El sistema debe poseer un buen rendimiento, lo cual se encuentra estrechamente vinculado a los requerimientos de hardware y software del sistema, los que se detallan a continuación. Debe tener un tiempo de respuesta entre 0 y 2 segundos.

Hardware

2 Procesadores Intel Dual Core o Core 2 Duo a 3.0 Ghz mínimo.
4 GB de memoria RAM.
100 GB de espacio disponible en disco.
Conexión de red Ethernet.

Software

Sistema Operativo: Debian Lenny o Ubuntu Server
Compilador GCC para C/C++
Servidor Web Apache 2.2
Servidor de Base de Datos: PostgreSQL
Lenguaje de Programación PHP 5

Interfaz

Interfaz simple, intuitiva y agradable al usuario. No hacer uso de más de 3 tonalidades de colores. Textos visibles y con buena tipografía.

Ayuda y documentación en línea

La ayuda y la documentación en línea deben estar en formato HTML, accesible a todos los usuarios del sistema.

2.4 Casos de uso del sistema

Para satisfacer los requisitos que debe cumplir el sistema, estos se agruparon en un total de 6 casos de uso, los cuales se detallan a continuación:

2.4.1 Gestionar Búsqueda

El caso de uso Gestionar Búsqueda tiene como finalidad la recuperación de los documentos almacenados en la base de datos del sistema, los cuales serán devueltos al actor que inicia el caso de uso en orden de relevancia de la información, de acuerdo al criterio de búsqueda introducido por el actor. El actor del sistema puede inicializar el caso de uso de diversas formas: mediante una búsqueda genérica en toda la web, incluyendo todos los tipos de archivos indizados en la base de datos, tales como imágenes y archivos de textos.

También puede inicializar el caso de uso mediante la búsqueda solo de imágenes, en este caso, el sistema solo recuperará aquellos resultados que coincidan con alguno de los archivos de imágenes más comunes, como son: .jpg, .gif, .png, entre otros. Mediante 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.

Durante la ejecución del caso de uso, el sistema obtiene todas las configuraciones existentes en el archivo de configuración *mnogosearch.yml.* Dichas configuraciones afectan el comportamiento del motor de búsqueda.

Otro aspecto de relevancia en el caso de uso, lo constituye la funcionalidad *sugerir palabra correcta*, la cual consiste en que si el usuario teclea incorrectamente el criterio de búsqueda, el sistema sugiere una posible palabra correcta para dicho criterio de búsqueda. Esta funcionalidad se puede configurar en el archivo de configuraciones mnogosearch.yml descrito con anterioridad.

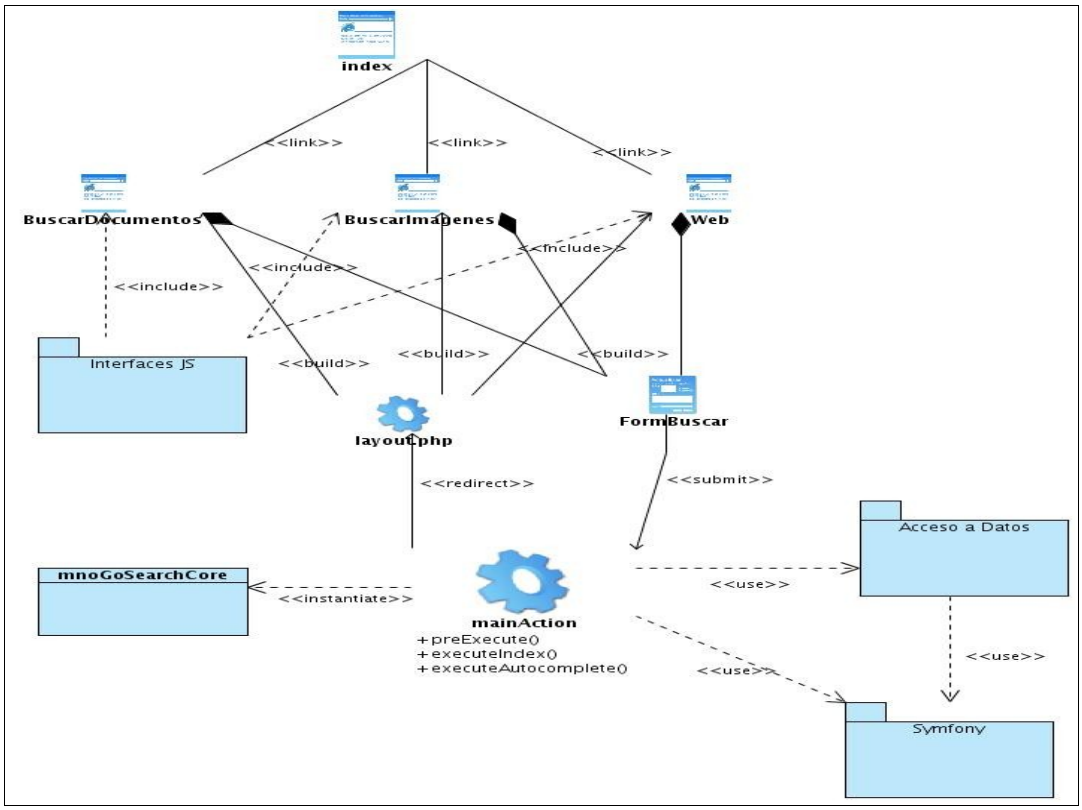

Figura 6. Diagrama de clases con estereotipos web del caso de uso Gestionar Búsqueda.

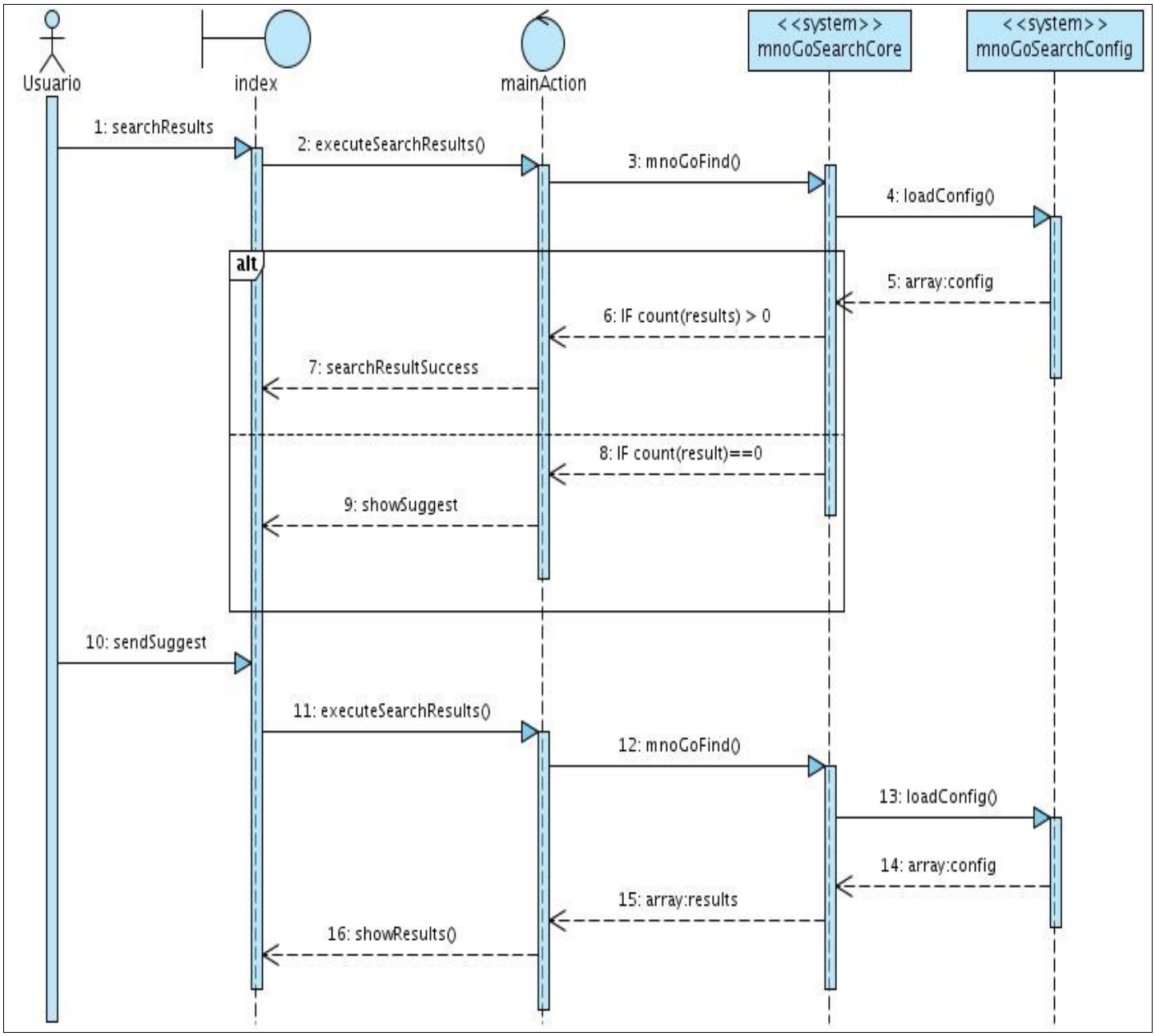

Figura 7. Diagrama de secuencias del caso de uso Gestionar Búsqueda.

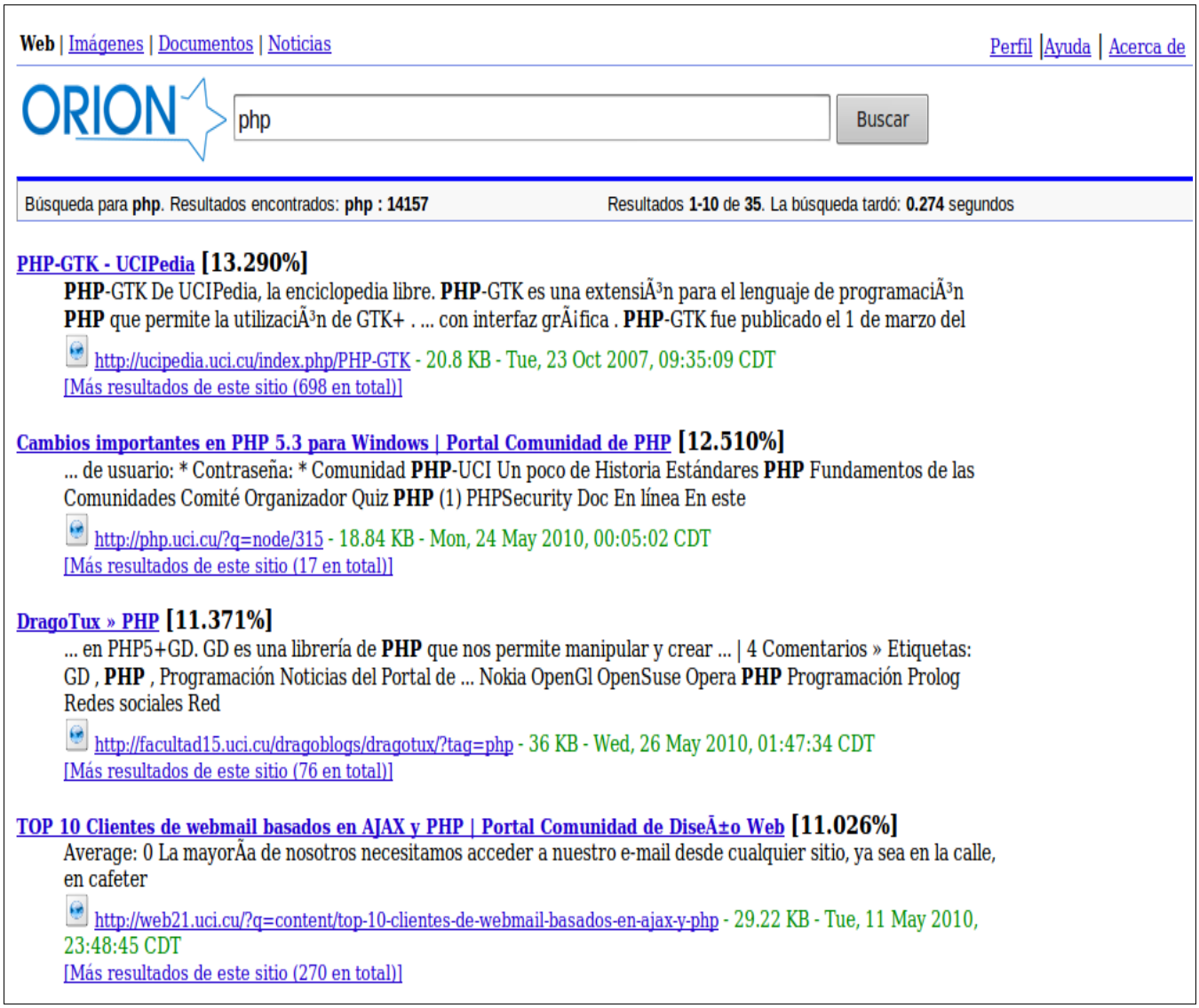

Figura 8. Vista del caso de uso Gestionar Búsqueda.

2.4.2 Gestionar Noticias

Mediante una tarea repetitiva con una frecuencia de aproximadamente 15 minutos, se obtienen todos los contenidos existentes en los canales RSS de la institución y los cuales deben estar almacenados en la base de datos del sistema. Dichos contenidos son almacenados en la base de datos para su posterior consulta. Una vez que el actor del sistema accede al mismo, debe tener acceso a todas las noticias almacenadas en la base de datos del sistema. Dichas noticias pueden ser calificadas positiva o negativamente por el usuario, lo cual incide directamente en el orden en que se muestran las noticias destacadas. También influye en la relevancia de las noticias la cantidad de veces que ha sido leída la misma. El sistema brinda la posibilidad de generar en formato PDF un boletín de noticias en un rango de tiempo determinado: día actual, ayer, la semana actual y el mes actual.

2.4.3 Gestionar Perfiles

Este caso de uso permite al actor gestionar toda la información de su perfil, el cual es creado la primera vez que accede al motor de búsqueda mediante el sistema de autenticación. Una vez creado el perfil, es posible gestionar los bookmarks de resultados obtenidos en búsquedas realizadas por el usuario. Como requisito previo a la ejecución de este caso de uso, es necesario haber ejecutado el caso de uso autenticar usuario.

Para ver diagrama de clases con estereotipos web, diagrama de secuencia y vista del sistema, remítase a los anexos 1, 2 y 3 respectivamente.

2.4.4 Gestionar Bookmarks

El actor del sistema, una vez que realice una búsqueda en el mismo, podrá almacenar aquellos resultados que resulten interesantes para su investigación. Estos resultados serán guardados en el perfil de cada usuario previamente autenticado en el sistema. Podrán ser editados o eliminados del perfil en cualquier

momento, previa autenticación del usuario en el sistema. Como requisito previo a la ejecución del caso de uso es necesario haber ejecutado el caso de uso autenticar usuario.

Para ver diagrama de clases con estereotipos web, diagrama de secuencia y vista del sistema, remítase a los anexos 1, 2 y 3 respectivamente.

2.4.5 Autenticar Usuario

El actor, en caso de que desee un motor de búsqueda personalizado, podrá autenticarse en el sistema mediante su propio usuario y contraseña del dominio de la universidad, en este caso, el dominio uci.cu. Este caso de uso chequea que las credenciales de acceso al sistema sean las correctas. De resultar positivo, se crea un perfil de usuario la primera vez que accede al sistema.
Para ver diagrama de clases con estereotipos web, diagrama de secuencia y vista del sistema, remítase al anexo 1, 2 y 3 respectivamente.

2.4.6 Enviar Correo

Este caso de uso permite enviar por correo electrónico el enlace a un recurso existente en la red de la universidad y el cual fue encontrado por el usuario mediante una consulta realizada al motor de búsqueda.

Para ver diagrama de clases con estereotipos web, diagrama de secuencia y vista del sistema, remítase al anexo 1, 2 y 3 respectivamente.

2.5 Diseño del sistema

En el desarrollo del sistema se utilizaron un grupo de plugins de Symfony previamente implementados por terceros y los cuales daban solución a distintas funcionalidades importantes para el sistema. A

continuación se describen brevemente los plugins de Symfony utilizados:

sfFeed2Plugin: plugin que permite crear canales RSS partiendo de los datos contenidos en una base de datos. Permite además, obtener los elementos existentes en los canales RSS en sus diferentes versiones, fundamentalmente la versión 1.0 y 2.0 del protocolo RSS.
RSS son las siglas de **Rich Site Summary** o **Really Simple Syndication**, y está diseñado para la distribución de noticias o información tipo noticias contenidas en sitios web y weblogs.

SfJqueryReloadedPlugin: Plugin que permite la integración de la librería de JavaScript Jquery en el framework Symfony.
jQuery es una biblioteca o framework de Javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el arbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

SfLightboxPlugin: plugin que integra Lightbox2 en el framework Symfony. Lightbox2 es una librería de JavaScript para adicionar útiles efectos en el trabajo con imágenes.

SfTCPDFPlugin: plugin que integra en Symfony la librería de PHP TCPDF, librería ampliamente utilizada en la generación de contenidos en formato PDF.

SfWebBrowserPlugin: integra un potente navegador web escrito en PHP. Muy útil para realizar peticiones web a servidores distantes sin la intervención del usuario. Es capaz de detectar los distintos códigos de estados y cabeceras HTTP devueltas por el servidor web.

SfMnogosearchPlugin: encapsula la lógica del motor de búsqueda mnoGoSearch, haciendo más fácil e intuitivo el desarrollo de aplicaciones basadas en el mismo. Para la correcta utilización del plugin, es necesario que esté previamente instaladas y configuradas las librerías de mnoGoSearch en el sistema.

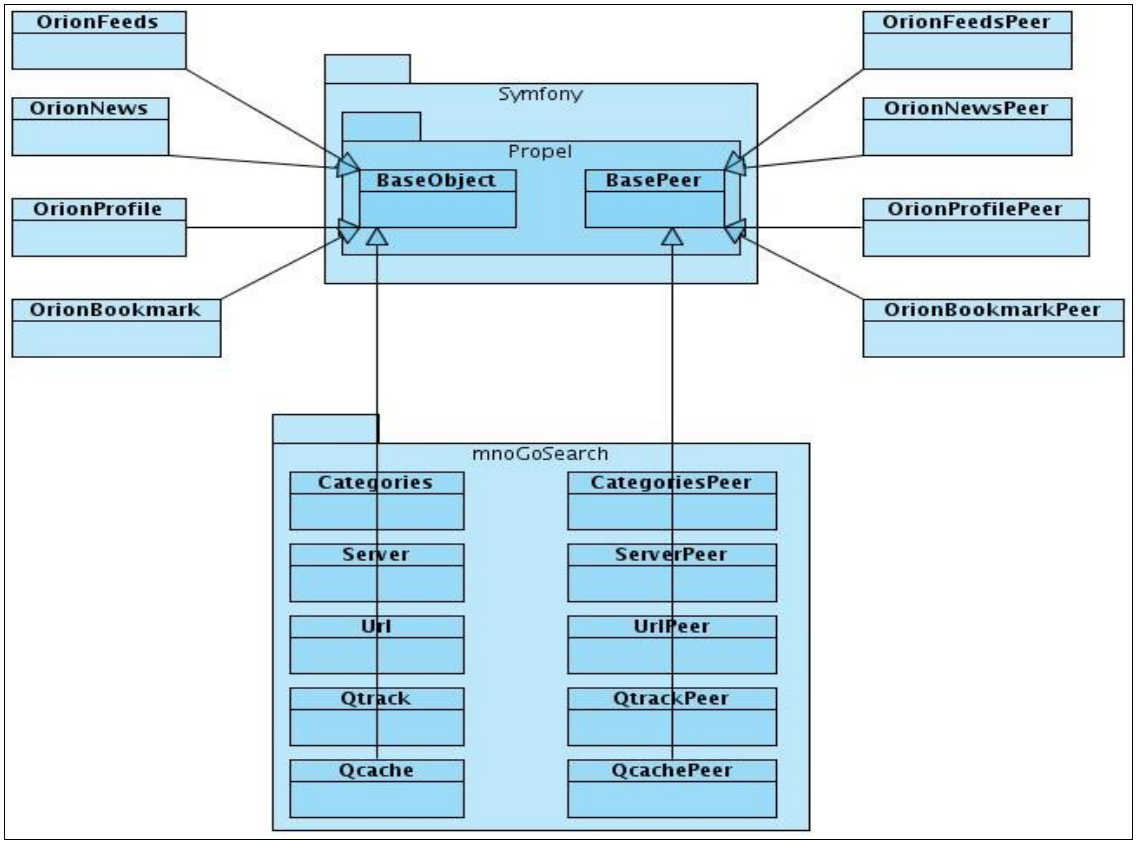

Figura 9. Diagrama de clases del sistema.

2.6 Diseño de la Base de Datos

La base de datos del sistema está compuesta por tablas que manipula directamente el spider utilizado y las tablas que almacenan los datos manipulados por el sistema propuesto. Las tablas del sistema propuesto, tienen el prefijo Orión para una mejor diferenciación entre las tablas del spider y las tablas del sistema a implementar.

El sistema, está compuesto, en su primera versión, por un total de 18 tablas, de las cuales, 4 fueron

definidas para uso del sistema. Estas tablas y sus atributos se describen a continuación:

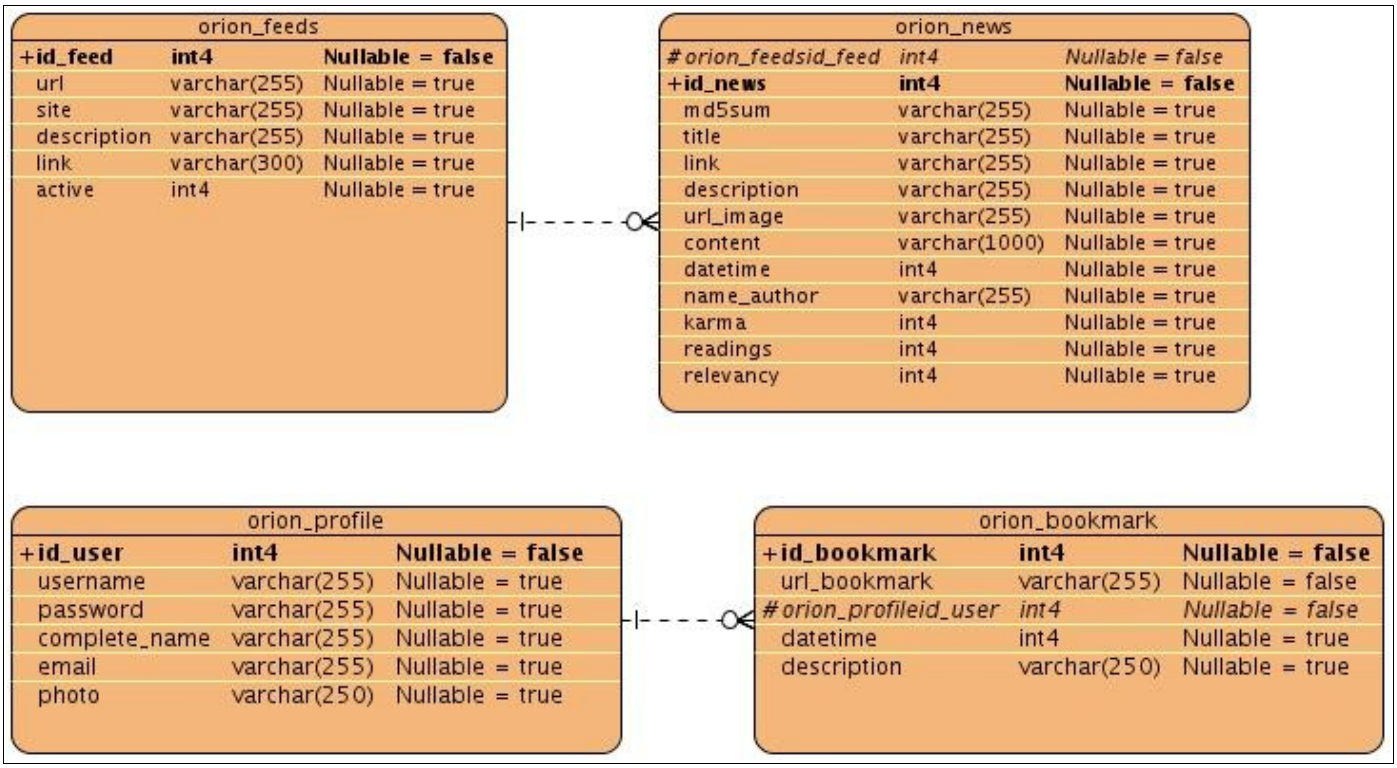

Figura 10. Modelo de datos del sistema.

Descripción de las tablas:

Nombre: orion_feeds		
Descripción: tabla que almacena los canales RSS desde los cuales se obtendrán las noticias.		
Atributo	**Tipo**	**Descripción**
id_feed	Integer	Es la llave primaría de la tabla. Valor autoincrementable que identifica a un único canal

		RSS dentro del sistema.
url	varchar	Url del canal RSS.
site	varchar	Sitio desde el cual se obtiene el canal RSS.
descripction	varchar	Breve descripción sobre el contenido del canal RSS
link	varchar	Url del sitio al que pertenece el canal RSS
active	integer	Entero que indica si el canal RSS está activo o no. Puede tomar uno de dos valores: 0 ó 1.

Tabla 1: Descripción de la tabla orion_feeds

Nombre: orion_news		
Descripción: tabla que almacena las noticias y contenidos obtenidos de los canales RSS descritos anteriormente.		
Atributo	**Tipo**	**Descripción**
id_news	integer	Llave primaria de la tabla. Representa a un único elemento dentro del sistema.
md5sum	varchar	Cadena que representa el código hash del algoritmo MD5 sobre el título de la noticia. Su objetivo es poder comprobar la unicidad de la noticia dentro del sistema.
title	varchar	Título de la noticia introducida al sistema y que es obtenida de los canales RSS.
link	varchar	Enlace de la noticia mediante el cual el actor del sistema podrá acceder a la misma.
description	varchar	Breve descripción de la noticia.
url_image	varchar	Url de la imagen de la noticia.
content	varchar	Contenido de la noticia si está disponible.
datetime	integer	Fecha y hora en la que se publicó la noticia.

name_author	varchar	Nombre completo del autor de la noticia si está disponible.
karma	integer	Número que representa el voto otorgado por los usuarios del sistema. Es empleado para calcular la relevancia de la noticia.
readings	integer	Cantidad de lecturas de la noticia en cuestión. Es utilizado para calcular la relevancia.
relevancy	integer	Número que representa la relevancia de la noticia. No es más que la suma de la cantidad de lecturas de la noticia más el karma.

Tabla 2: Descripción de la tabla orion_news.

Nombre: orion_profile		
Descripción: tabla que almacena los perfiles de los usuarios que han accedido al sistema.		
Atributo	**Tipo**	**Descripción**
id_user	integer	Identificador único de usuario. Es la llave primaria de la tabla.
username	varchar	Nombre de usuario del sistema. Generalmente es obtenido del servidor LDAP.
password	varchar	Contraseña de acceso al sistema para cada uno de los usuarios registrados.
complete_name	varchar	Nombre completo del usuario registrado en el sistema.
email	varchar	Dirección de correo electrónico del usuario registrado en el sistema.
photo	varchar	Dirección de la foto del usuario del sistema.

Tabla 3: Descripción de la tabla orion_profile.

Nombre: orion_bookmark		
Descripción: tabla que almacena los bookmarks de los resultados encontrados por los usuarios en el motor de búsqueda.		
Atributo	**Tipo**	**Descripción**
id_bookmark	integer	Identificador único del bookmark en el sistema.
url_bookmark	varchar	URL del bookmark a guardar en el perfil.
datetime	integer	Fecha y hora en que se guardó el bookmark.
description	varchar	Breve reseña del bookmark.

Tabla 4: Descripción de la tabla orion_bookmark

2.7 Implementación del sistema

Durante el desarrollo del sistema, se implementó y/o reutilizó los componentes de software que se describen a continuación:

- mnoGoSearch: constituye las librerías de mnoGoSearch para el lenguaje de programación PHP 5. Dicha librería contiene las principales funciones utilizadas en la interacción con las opciones que brinda el software mnoGoSearch.
- Symfony: constituyen las librerías utilizadas del framework symfony.
- Plugins: paquete de plugins implementados por terceros que sirvieron de base para la implementación de algunas funcionalidades necesarias para el sistema.
- Orion: constituye las clases y estructuras de datos del sistema implementado.

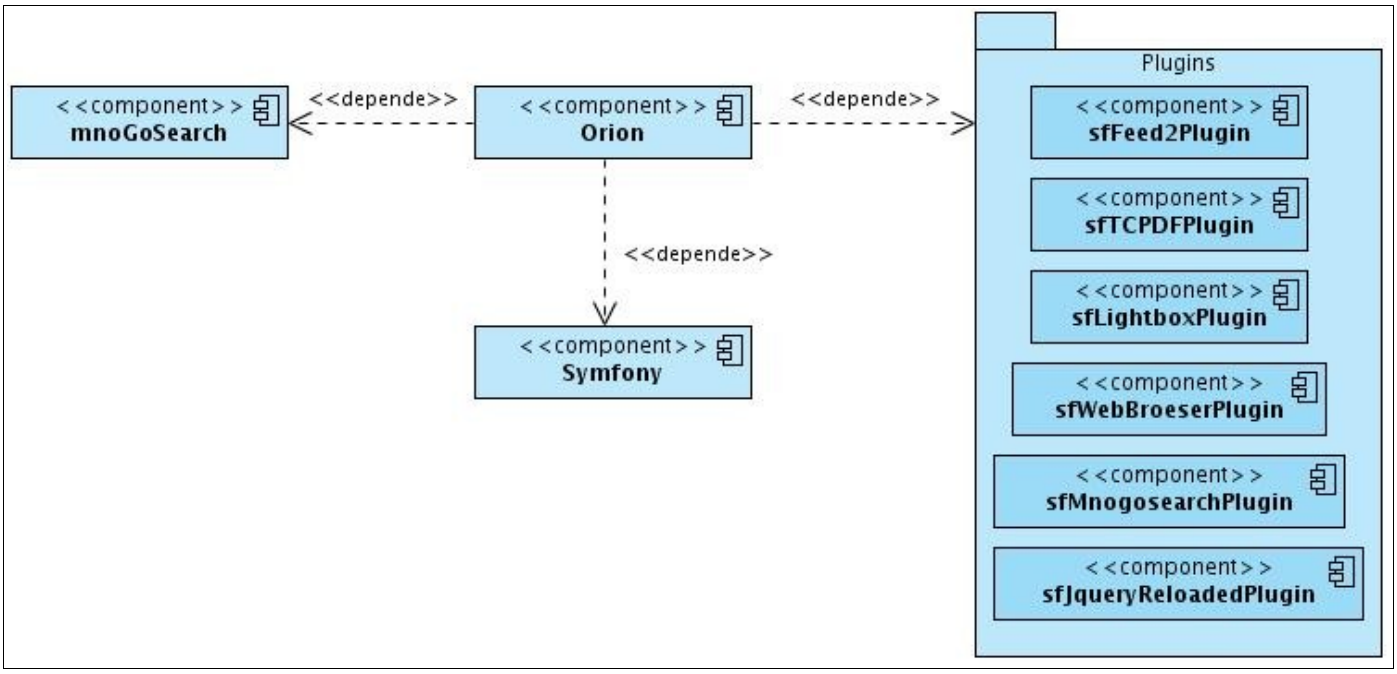

Figura 11. Diagrama de componentes del sistema.

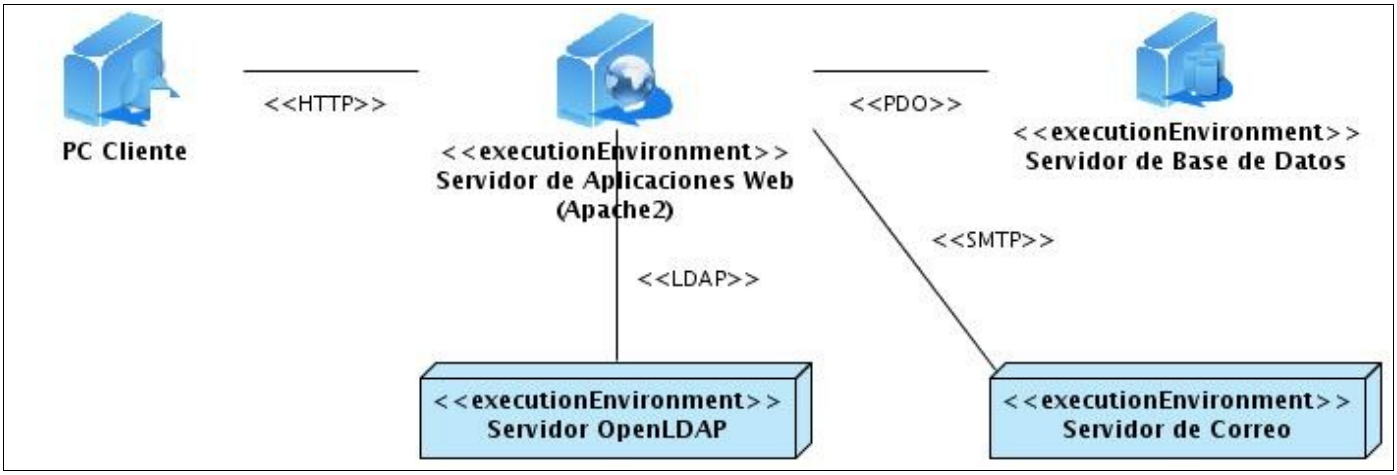

Figura 12. Diagrama de despliegue del sistema.

2.8 Conclusiones parciales

Durante el diseño e implementación del sistema, se logró el modelamiento y programación de las clases y estructuras de datos fundamentales sobre las que se continuará el desarrollo del sistema. Se logró además, la obtención de un modelo de datos que satisface las principales necesidades de almacenamiento que requiere el sistema, así como los principales nodos y sus protocolos de comunicación empleados en el despliegue del mismo.

Capítulo 3. Evaluación de la solución implementada

3.1 Introducción

La evaluación de los Sistemas de Recuperación de Información no es nueva. Muchos investigadores destacados en el tema han aportado a esta complicada ciencia. Los primeros estudios sobre el tema, tuvieron lugar en el Instituto Cranfield de Tecnologías, en el año 1957. Son dos los estudios Cranfield más importantes, el primero de ellos, tenía como objetivo 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[38]. En dicha metodología, las métricas son agrupadas en tres grandes grupos: las métricas técnicas, de calidad y orientadas a la persona. En la evaluación de la solución implementada se utilizaron dicho conjunto de métricas definidas en dicha investigación.

3.2 Métricas técnicas

3.2.1 Composición de los Índices

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

En este aspecto se destacan tres componentes importantes: Tamaño del índice, Frecuencia de Actualización y Porción de página web indexada (título, primeros párrafos, página completa, etiquetas meta). Las magnitudes de cada motor dependerán del hardware y el software dedicado.

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

El tamaño del índice o cubrimiento del Sistema de Recuperación de Información, es calculado aplicando la siguiente fórmula:

$$Cubrimiento = \frac{Páginas\ Indizadas}{Total\ de\ Páginas}$$

Donde 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 esta 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, sus contenidos no llegan a ser indizados 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 en el rendimiento del spider en el proceso de descubrimiento de la información y en la calidad en los índices. En el caso nuestro, se realizó 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 indizada

La porción de la página web indizada es definida mediante el sistema de configuración del spider. En dicho sistema de configuración, se definió un grupo de secciones dentro de la página web con sus respectos pesos, teniendo el 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. Dicho peso influye en el orden de relevancia con que se muestran los resultados obtenidos en el motor de búsqueda.
Sección body: el 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.

Note que 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 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 sistemas necesitan 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 Capacidades y sintaxis de las consultas

El motor de búsqueda, al ser totalmente configurable, brinda la posibilidad de seleccionar el modo de búsqueda a utilizar. Existen cuatro modos de búsquedas fundamentales:

- **UDM_MODE_ALL:** en este modo de búsqueda, todas las palabras que conformar la consulta introducida por el usuario son tomados en cuenta y por tanto, todas las palabras serán buscadas en la base de datos en forma independiente.
- **UDM_MODE_ANY:** en este modo de búsqueda se recuperarán y mostraran al usuario, todos aquellos documentos que contengan en su estructura, alguna de las palabras que forman la consulta, sin importar si todas las palabras aparecen en el mismo documento.
- **UDM_MODE_PHRASE:** en este modo de búsqueda, el criterio introducido por el usuario será

tomado en cuenta como un todo, es decir, los documentos encontrados en los índices, contendrán todas los vocablos que forman la consulta, tal y como se introdujo en el sistema.

– **UDM_MODE_BOOL:** en este modo, se intentará recuperar los documentos haciendo uso de los operadores booleanos AND, OR, NOT, los cuales permiten realizar consultas complejas al motor de búsqueda.

3.2.4 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 y Excriba, orientadas a la producción de software en nuestra universidad.
– **blogs:** los blogs temáticos constituyen los sitios web con más aceptación dentro de la universidad.
– **docencia:** en esta categoría entrarían las plataformas de teleformación de la sede principal, las plataformas de las facultades regionales y la plataforma del postgraduado.
– **facultades:** en esta categoría se incluyen las intranets de las facultades regionales y la de la sede principal.

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

3.2.5 Interfaz y accesibilidad al buscador

El motor de búsqueda cuenta con una interfaz web sencilla que permite acceder a los contenidos en muy pocos clic de profundidad. Se puede considerar un sistema intuitivo y fácil de utilizar, en el cual se ha intentado cuidar su apariencia, acercándolo más al parecido con otros sistemas similares, evitando entrar en conflicto con la experiencia que posee el usuario en el uso de otros sistemas con mismos fines.

3.2.6 Servicios adicionales

El sistema cuenta con un servicio adicional de vital importancia para la comunidad universitaria. El servicio de noticias permite mantener un seguimiento sobre los principales canales RSS de la universidad, aglutinando en un solo lugar todos los contenidos noticiosos generados dentro de la universidad. Este servicio, proporciona además, una manera fácil de generar boletines noticiosos en formato PDF.

Indicador	Evaluación
Tamaño del índice	90%
Frecuencia de actualización del índice	1, 3, 7 y 15 días.
Porción de la página web indizada	95,00%
Tiempos de respuestas	Entre 0 y 2 segundos
Capacidad y sintaxis de las consultas	4 modos disponibles
Especialización en materias	Agrupamiento por categorías o etiquetas
Interfaz y accesibilidad al buscador	Buena
Servicios adicionales	Servicio de noticias

Tabla 5: Indicadores evaluados en prueba piloto.

3.3 Métricas de calidad

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 llevan 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 dependerá, 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* y *descripction*. 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.

Indicador\Consulta	Java	Firefox	MVC	Drupal	PHP
Enlaces relevantes	7	8	6	9	10
Enlaces muertos	0	0	0	0	0
Calidad de los resúmenes	Regular	Bien	Regular	Bien	Regular

Tabla 6. 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 sobre el mismo, en aras de detectar posibles deficiencias en los algoritmos empleados y en las características de la web analizada. En dichas pruebas, se obtuvo un grupo de deficiencias que se muestran a continuación:

Se detectó la existencia de algunos problemas con el posicionamiento web para buscadores, tema conocido como SEO[8].

Sitio\Etiqueta	Título	Descripción	Palabras claves	Posición
Octavitos	Si	Si	Si	1
Biblioteca	No	No	No	3
Firefoxmania	Si	No	No	2
Primavera	Si	Si	Si	1
Software Libre	Si	Si	No	17
Intranet	No	No	No	16
Facultad15	Si	Si	Si	1

Tabla 7. Comportamiento del posicionamiento web en algunos sitios de la red.

.

También se detectó una baja actualización de los enlaces existentes en los sitios web, lo que provoca que

8 SEO: *Search Engine Optimization u Optimización para motores de búsqueda.*

dichos enlaces no lleven al usuario a ningún lugar dentro de la web. Estos enlaces también son conocidos como enlaces muertos o enlaces rotos. De un total de 309 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.

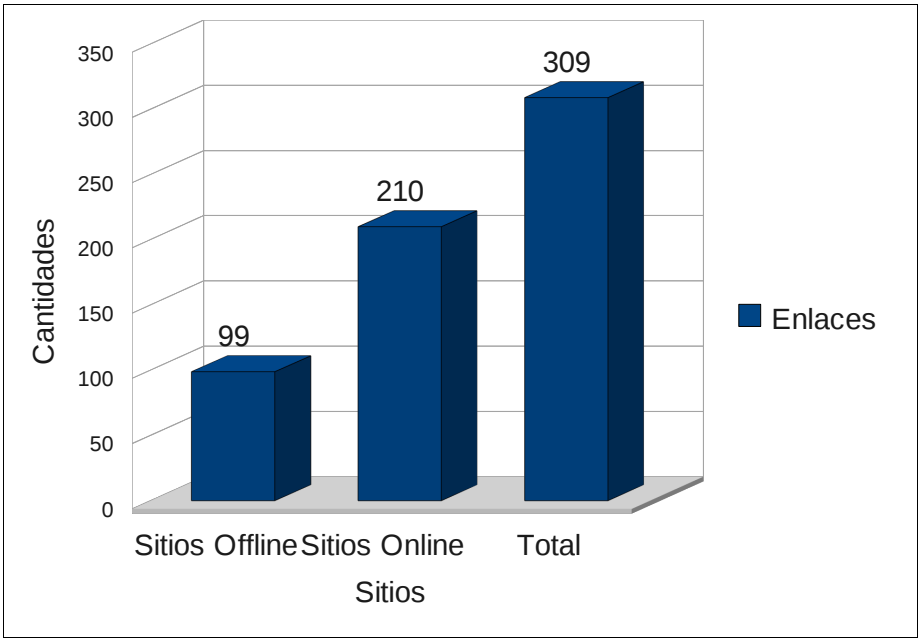

Figura 13. Gráfica que ilustra el estado de los enlaces existentes en la web de la red interna.

3.4 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 Universidad de las Ciencias Informáticas. ” con la cual se obtuvo muy buenos resultados en su aplicación. Se logró además la identificación de las fortalezas y debilidades que presenta el sistema desarrollado, además de un grupo de 2 deficiencias existentes en la web de la Universidad.

Conclusiones

Luego de estudiados varios sistemas de recuperación de información, se optó por utilizar el buscador mnoGoSearch.

Se definió una arquitectura para el nuevo sistema a desarrollar que brinda robustez y flexibilidad para el desarrollo de los principales requisitos de software identificados.

Se obtuvieron los artefactos necesarios para el desarrollo del sistema, aplicando RUP como metodología de desarrollo de software y UML como lenguaje de modelado.

Se logró la obtención de un producto de software con las características y requisitos identificados, el cual se espera, incremente la optimización de la búsqueda y recuperación de la información existente en la red de la Universidad.

Se aplicó una metodología de evaluación al sistema desarrollado que arrojó muy buenos resultados en la identificación de las fortalezas y debilidades existentes en el sistema.

Se identificó un grupo de características y debilidades existentes en la web de la Universidad que resultan de vital importancia su conocimiento en aras de mejorar la calidad de la información existente en la Universidad.

Recomendaciones

Se recomienda continuar el desarrollo del motor de búsqueda, añadiendo nuevas y atractivas funcionalidades y servicios que permitan mejorar la experiencia del usuario. Se recomienda además, hacer de conocimiento público y a las personas y entidades involucradas, sobre la existencia de las debilidades detectadas en la web de la Universidad para su posterior análisis y corrección.

Referencias bibliográficas

[1] Trejo, Raúl, Vivir en la Sociedad de la Información, 2001 [Disponible en: http://www.oei.es/revistactsi/numero1/trejo.htm]

[2] Baeza-Yates, Ricardo , Página Web de Ricardo Baeza-Yates, [Disponible en: http://www.dcc.uchile.cl/~rbaeza/spanish.html]

[3] Baeza-Yates, Ricardo, Modern Information Retrieval, 1999

[4] Martínez Méndez, Francisco J., Propuesta y desarrollo de un modelo para la evaluación de la recuperación deinformación en internet, 2002

[5] Pinto, María, Búsqueda y Recuperación de Información, 2004 [Disponible en: http://www.mariapinto.es/e-coms/recu_infor.htm#ri1]

[6] López Herrera, Antonio G., Modelos de Sistemas de Recuperación de Información Lingüística Difusa, 2006

[7] Cruz Almaguer, José A., Buscador Web, 2004

[8] Aguirre, Jorge D., Arquitectura de un buscador, 2007 [Disponible en: http://buscadores.fullblog.com.ar/post/arquitectura_de_un_buscador_531191953898/]

[9] Kiva, Buscadores o Search Engine, 2009 [Disponible en: http://www.searchoptimization.es/buscadores-search-engines/buscadores-search-engines.htm]

[10] Consoft, ¿Qué son los metabuscadores?, 2002 [Disponible en: http://www.consoft.es/noticias/news_text.asp?id=33219]

[11] Palacios, Myra, ¿Qué son los FFA's?, 2003 [Disponible en: http://www.trucoswebmasters.com/a182-13.html]

[12] Ricciardi, Agustín, Buscadores Verticales, 2009 [Disponible en:

http://blog2puntocero.wordpress.com/2009/01/28/buscadores-verticales/]
[13] García Broncano, Rubén, Modelos de Recuperación de la Información, 2006 [Disponible en: http://modelosrecuperacion.tripod.com/]
[14] Ayala Pichardo, Julio A., Modelo de Recuperación Booleano, 2007 [Disponible en: http://recuperacioninf.orgfree.com/modelo_booleano.html]
[15] Pulido, Sergio Martín, Modelo de Recuperación Probabilístico, 2007 [Disponible en: http://modelosderecuperacioni.iespana.es/probabilistico.html]
[16] Martínez Comeche, Juan A., Los modelos clásicos de recuperación de información y su vigencia, 2008
[17] Figuerola, Carlos E., Modelos Teóricos de Recuperación de Información, 2008
[18] Suárez Molina, Jhonlier, Spider UCI, 2004
[19] Colectivo, Historia de Google, 2009 [Disponible en: http://www.cad.com.mx/historia_de_google.htm]
[20] Comin, Javier , La historia de Yahoo, 2001 [Disponible en: http://www.maestrosdelweb.com/editorial/yahoohis/]
[21] Group Htdig, Internet search engine software, 2005 [Disponible en: http://www.htdig.org]
[22] Swish Team, Swish-e :: Home Page , 2010 [Disponible en: http://swish-e.org/index.html]
[23] Nutch Team, About Nutch, 2009 [Disponible en: http://lucene.apache.org/nutch/about.html#Overview]
[24] Nioche, Julien, Features - Nutch Wiki, 2009 [Disponible en: http://wiki.apache.org/nutch/Features]
[25] Barkov, Alexander, mnoGoSearch 3.3.9 reference manual, 2009 [Disponible en: http://www.mnogosearch.org/doc33/msearch-intro.html#features]
[26] IBM, IBM - Rational Unified Process, 2009 [Disponible en: http://www-

01.ibm.com/software/awdtools/rup/]

[27] PHP Group, PHP: Hypertext Preprocessor, 2009 [Disponible en: http://php.net/]

[28] Rodas Hinostroza, Raul , Características de PHP, 2005 [Disponible en: http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP]

[29] Java Team, Conozca más sobre la tecnología Java, 2009 [Disponible en: http://www.java.com/es/about/]

[30] Potencier, Fabien, Symfony | Web PHP Framework, 2010 [Disponible en: http://www.symfony-project.org/]

[31] Zend Team, About Zend Framework, 2010 [Disponible en: http://framework.zend.com/about/overview]

[32] Kumbia Team, Que es Kumbia, 2009 [Disponible en: http://www.kumbiaphp.com/blog/about/]

[33] Doctrine Team, Doctrine ORM for PHP, 2010 [Disponible en: http://www.doctrine-project.org/documentation/manual/1_2/en]

[34] Team Propel, Propel ORM, 2009 [Disponible en: http://www.propelorm.org/]

[35] Martínez, Rafael, Principal | www.postgresql-es.org, 2010 [Disponible en: http://www.postgresql-es.org/principal]

[36] Oracle Corp., About MySQL, 2009 [Disponible en: http://www.mysql.com/about/]

[37] Avila Yusniel, Llanes Néstor, Sistema Automatizado para la gestión de información en rehabilitación, 2008

[38] Mireldis García del Valle, Metodología para la evaluación de Sistemas de Recuperación de Información Web en la Universidad de las Ciencias Informáticas, 2009

Anexos

Anexo 1: Diagramas de clases con estereotipos web

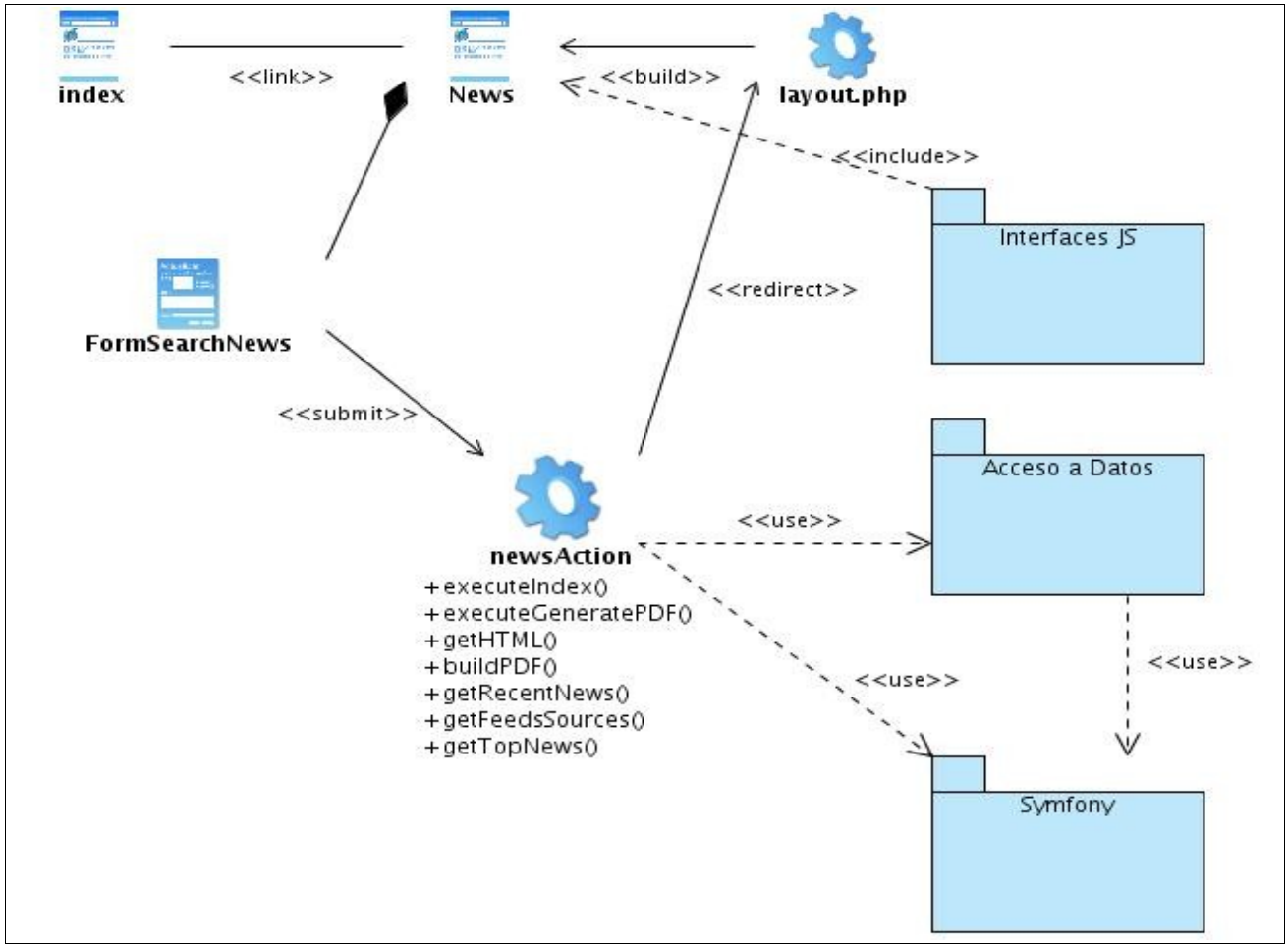

Figura 14. Diagrama de clases del caso de uso Gestionar Noticias.

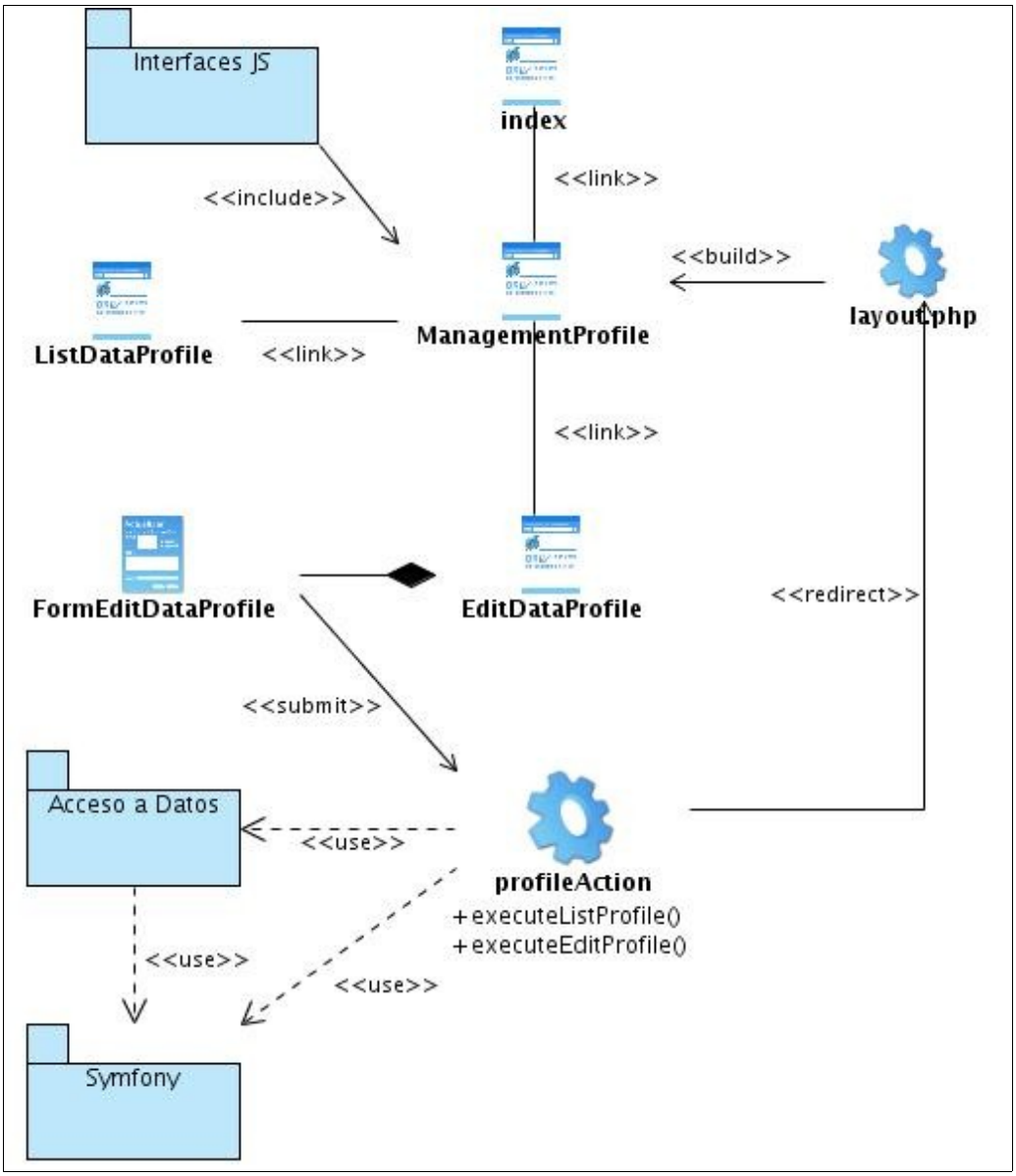

Figura 15. Diagrama de clases del caso de uso Gestionar Perfil.

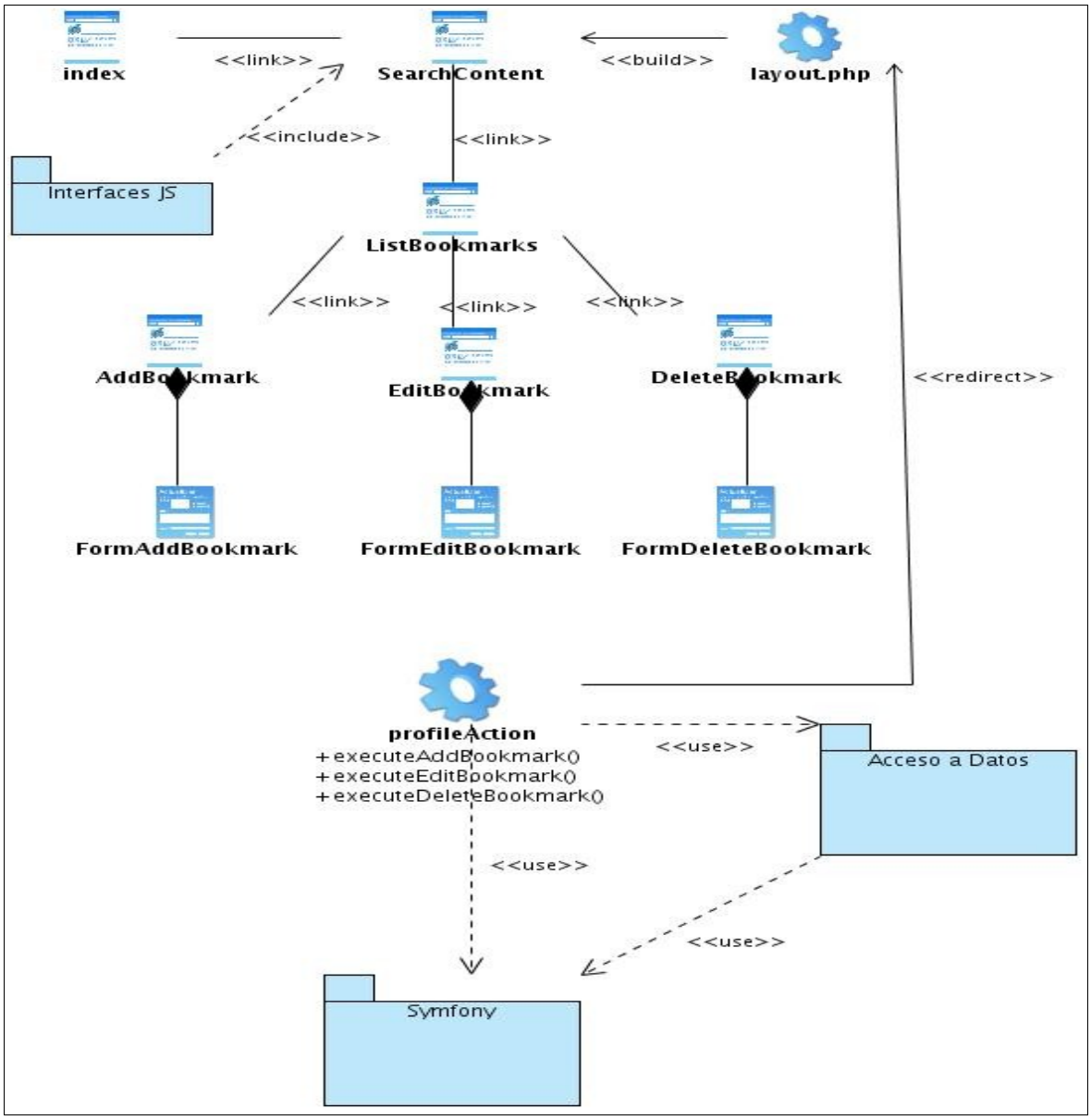

Figura 16. Diagrama de clases del caso de uso Gestionar Bookmars.

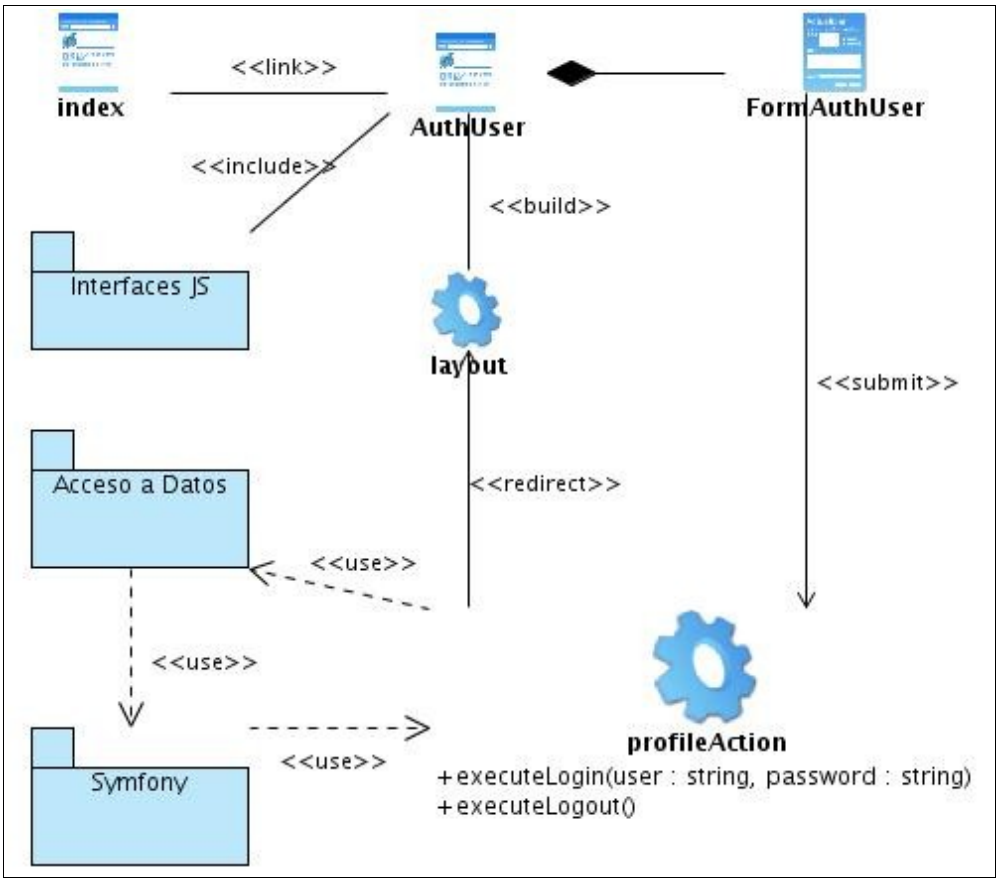

Figura 17. Diagrama de clases del caso de uso Autenticar Usuario.

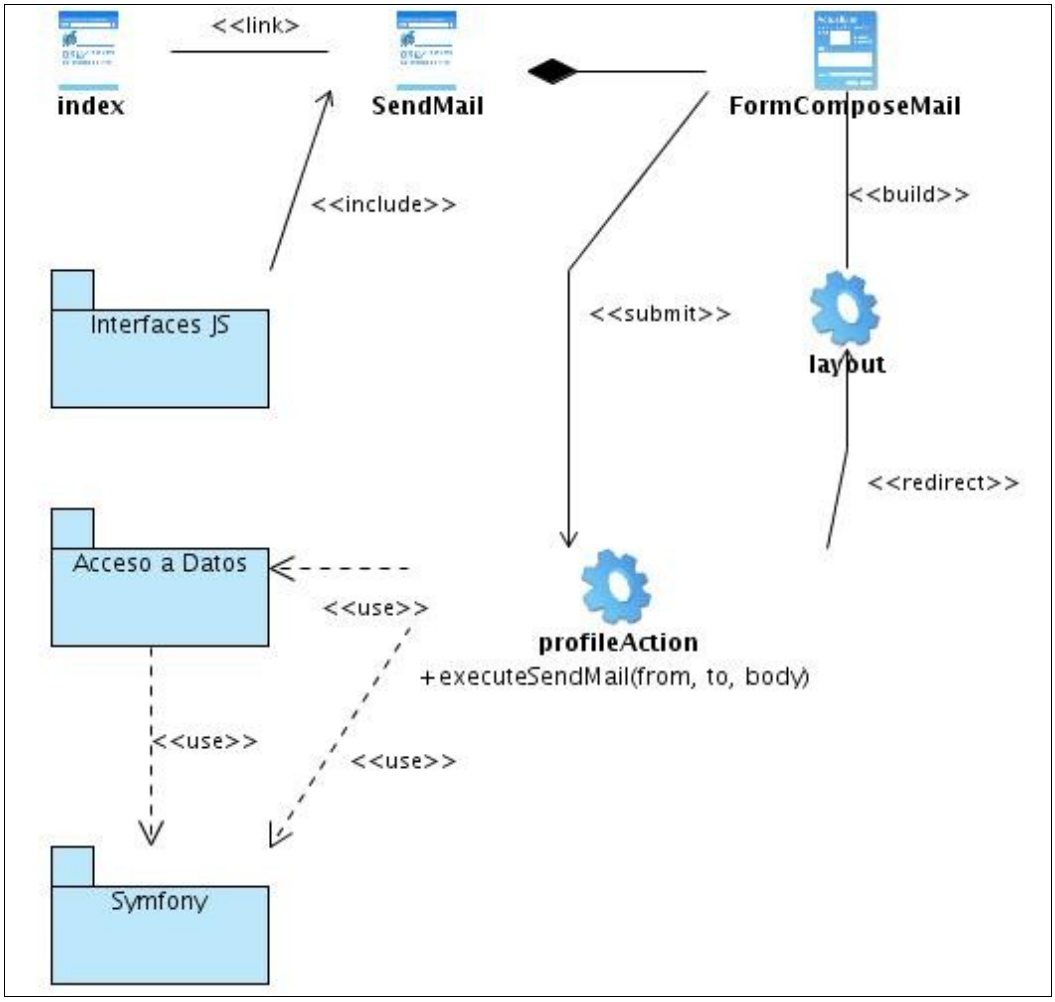

Figura 18. Diagrama de clases del caso de uso Enviar Correo.

Anexo 2: Diagramas de secuencia de los casos de uso.

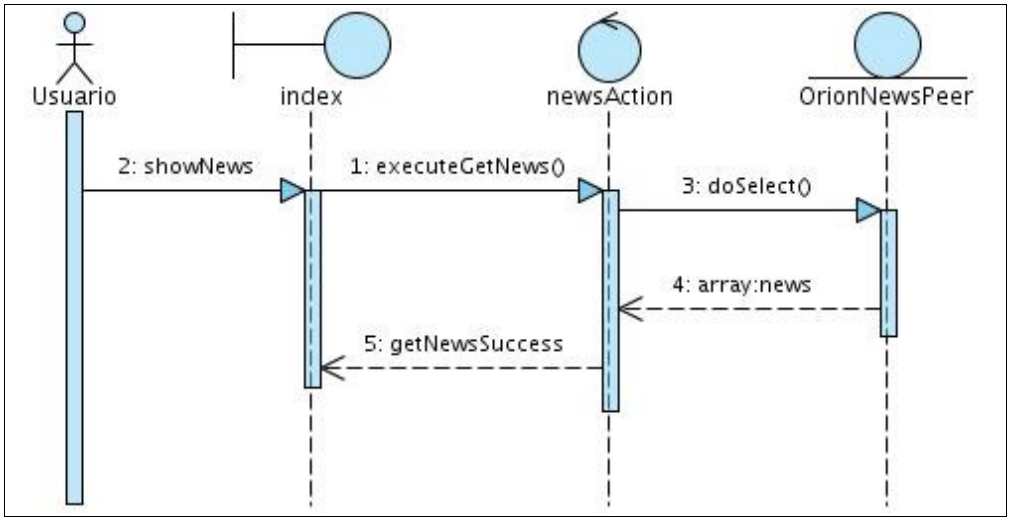

Figura 19. Diagrama de secuencia del caso de uso Gestionar Noticias

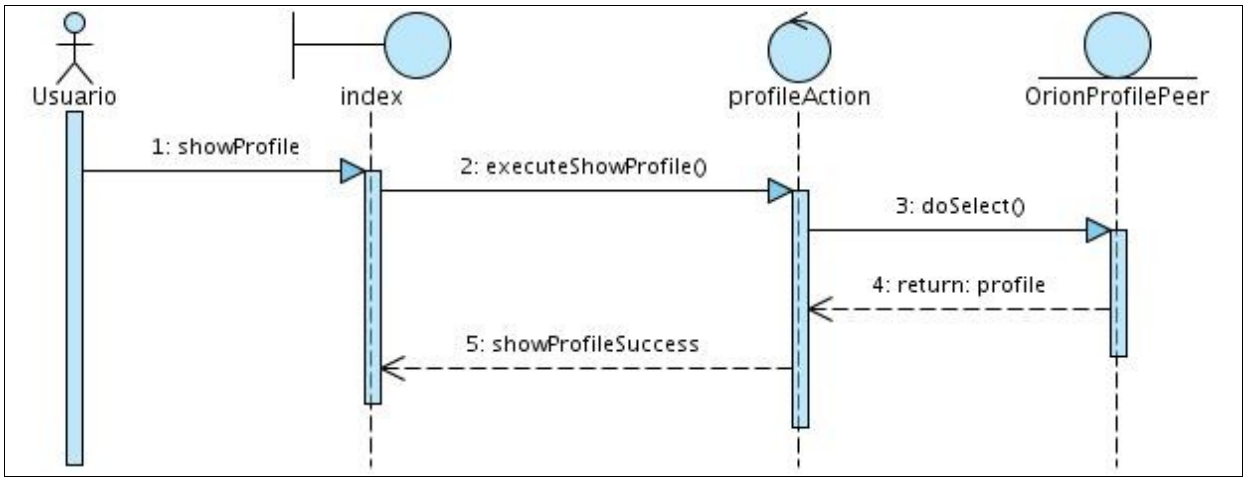

Figura 20. Diagrama de secuencia del caso de uso Gestionar Perfil.

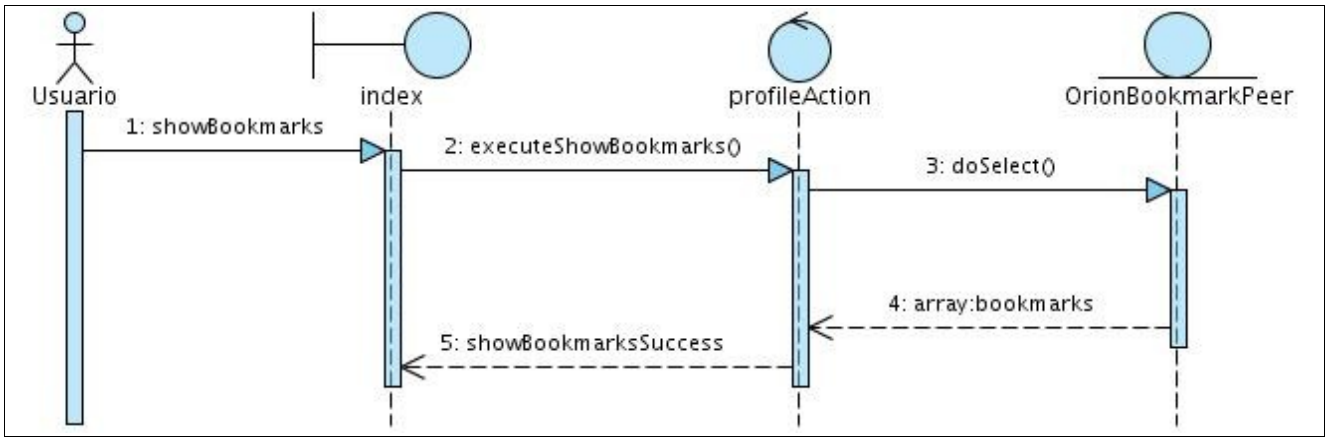

Figura 21. Diagrama de secuencia del caso de uso Gestionar Bookmarks.

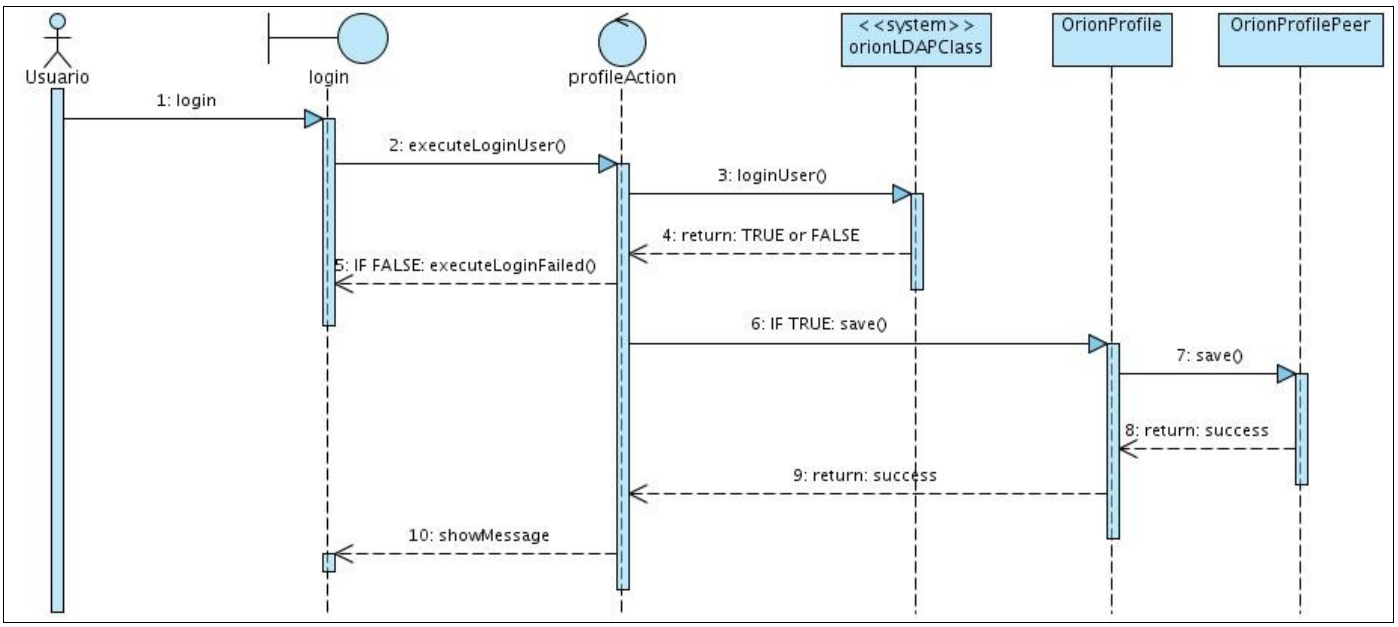

Figura 22. Diagrama de secuencia del caso de uso Autenticar Usuario.

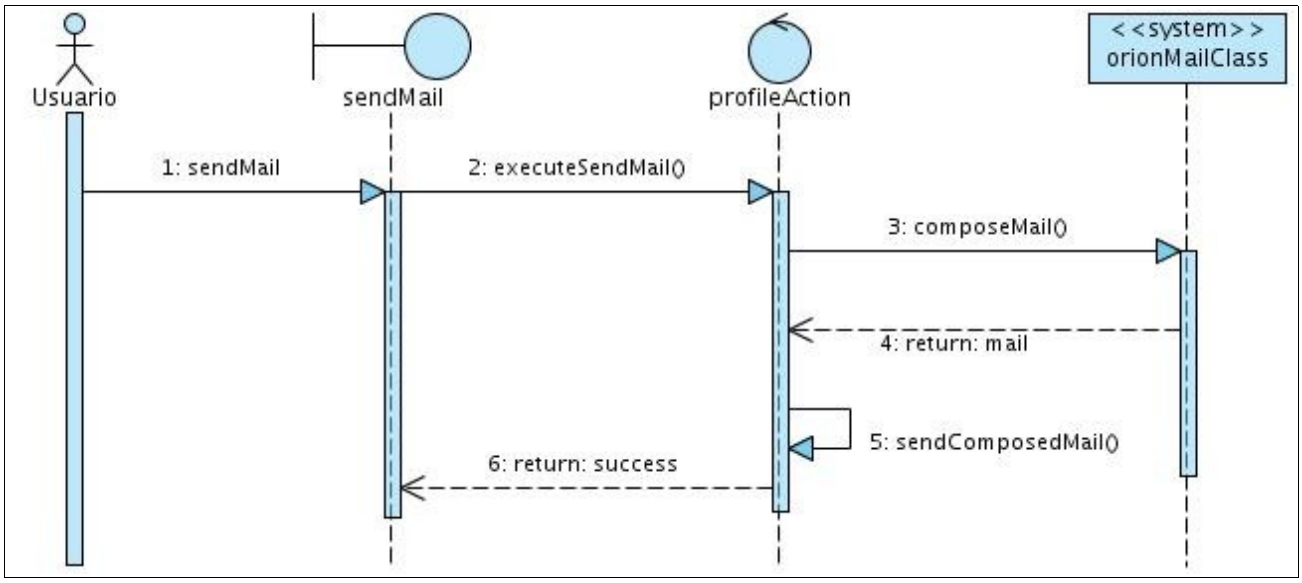

Figura 23. Diagrama de secuencia del caso de uso Enviar Correo.

Anexo 3: Vistas del sistema.

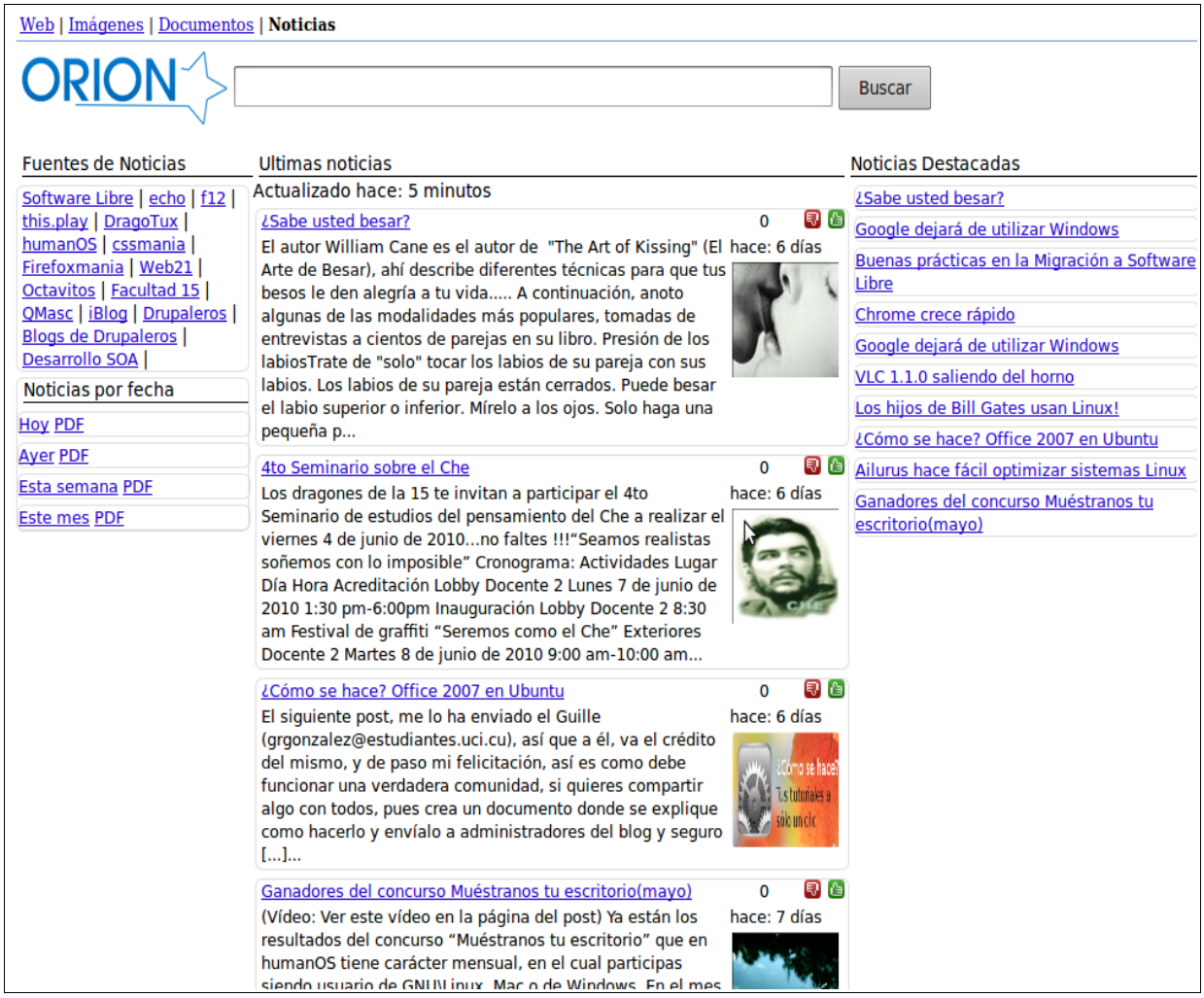

Figura 24. Vista del caso de uso Gestionar Noticias

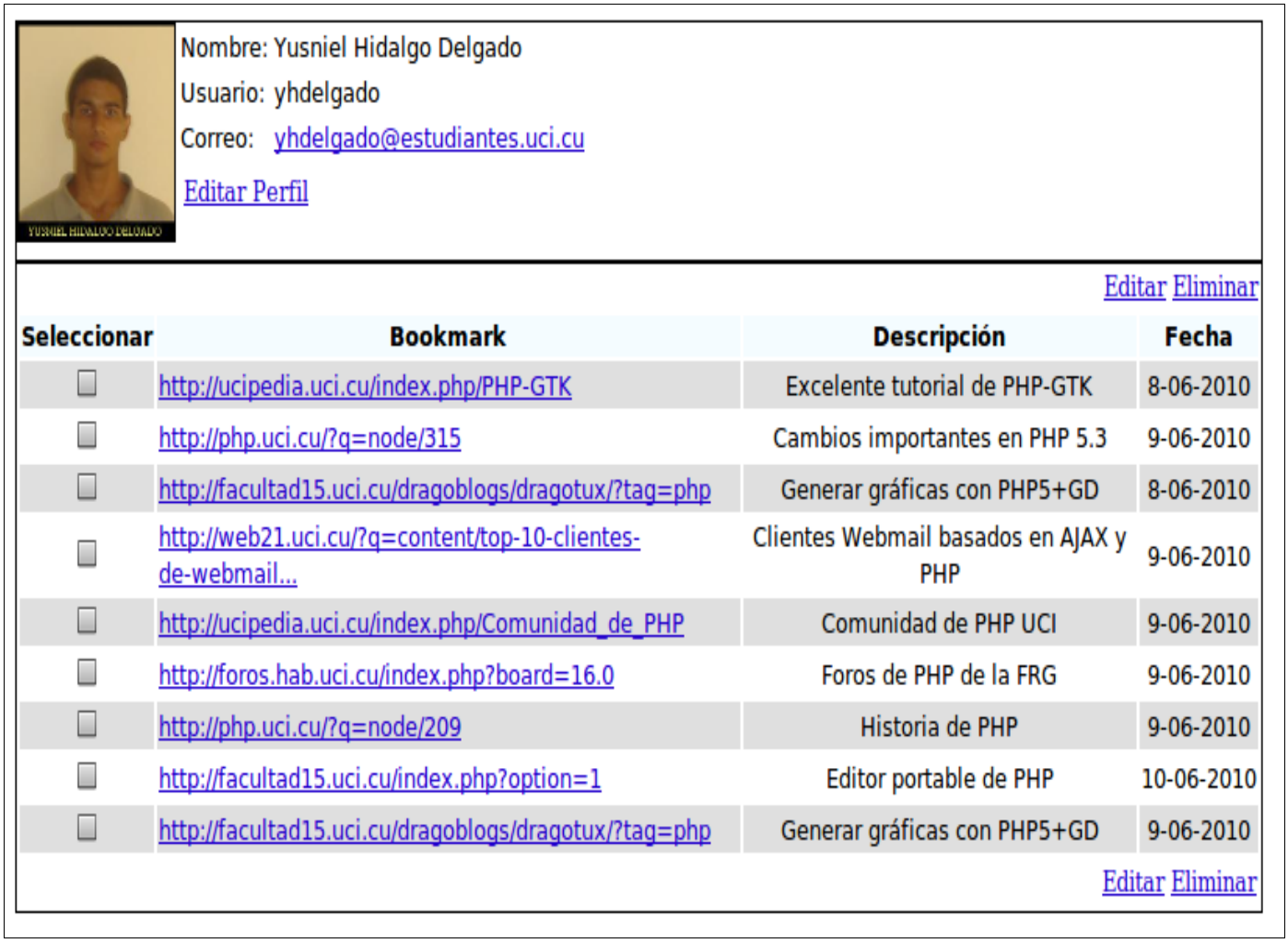

Figura 25. Vista de los casos de uso Gestionar Perfil y Gestionar Bookmarks.

Figura 26. Vista del caso de uso Autenticar Usuario.

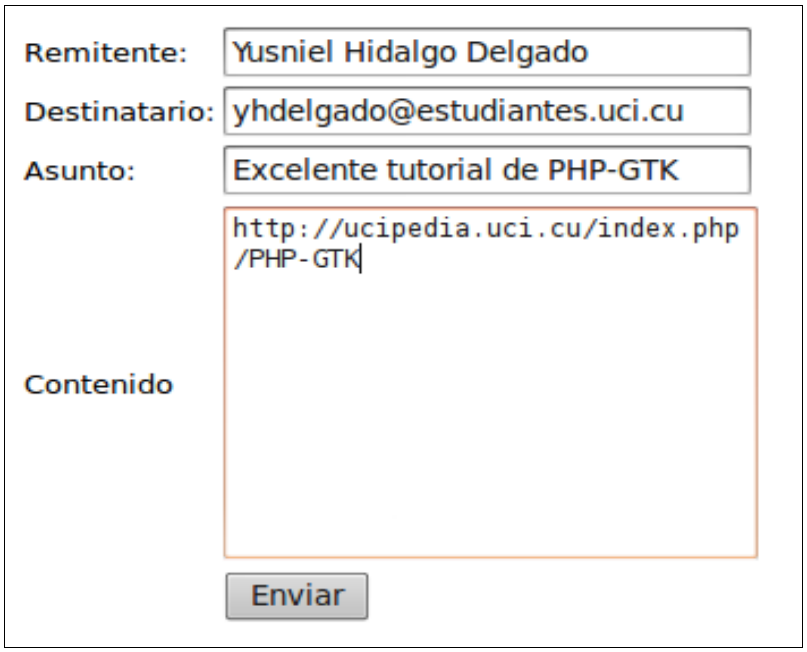

Figura 27. Vista del caso de uso Enviar Correo.