Observatorio de Internet para la ESPOL

E. Ponguillo, A. Ponguillo, F. Echeverría
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral
Campus Gustavo Galindo, Km. 30.5 Vía Perimetral, Apartado postal 09-01-5863, Guayaquil, Ecuador
eponguil@espol.edu.ec, aponguil@espol.edu.ec, pechever@espol.edu.ec

Resumen

La creación de un sitio web persigue alcanzar los objetivos e intereses de sus propietarios, cuya permanencia y éxito en la red dependen del posicionamiento que nos den los buscadores. El presente artículo presenta una herramienta WEB para la ESPOL que permite tener información de varios parámetros SEO (Search Engine Optimization) de los sitios WEB de las distintas universidades del país. La aplicación permite registrar la URL del sitio WEB de la universidad y, mediante la misma, se conecta a los servidores de Alexa y Yahoo! Mediante sus diferentes APIs, obtenemos la información requerida, la cual es almacenada en la base de datos para efectos de consultas a la misma. De igual forma, mediante consulta en el parámetro URL, extraemos el contenido de la página devuelta. Para lograr monitorear y actualizar la información, dentro de la implementación del sistema, se emplea un demonio que cada 24 horas lanza procesos para actualizar las diferentes métricas.

Palabras Claves: *SEO, Yahoo!, Alexa, API, WEB, spamdexing, demonio, buscador*

Abstract

Creating a web site intended to implement the goals and interests of its owners, whose tenure and success in the network depend on the position to give us the search engines. This article presents a web tool for ESPOL that allows take several parameters SEO (Search Engine Optimization) of the Web sites of various universities. The application can record the URL of the Web site of the university and why it connects to servers Alexa and Yahoo through its various APIs , we get the required information, which is stored in the database query effects to it. Similarly, using URL query parameter, extract the contents of the returned page. To achieve monitor and update information within the system implementation uses a daemon that processes launched every 24 hours to update different metrics.
.

Key Words: *SEO, Yahoo!, Alexa, API, WEB, spamdexing, daemon, search engine*

1. Introducción

El desarrollo de un sitio WEB no termina con la puesta en producción del mismo, sino que debe someterse a un constante monitoreo que determine, en un momento dado, el grado de interés que produce. El uso de estándares WEB no basta para tener una posición consolidada en internet. Hay que considerar para el efecto varios indicadores, que llevados adecuadamente, lograrán que los motores de búsquedas nos coloquen en las mejores posiciones en sus resultados.

Este artículo pretende dar una mirada a una de las grandes dificultades para lograrlo, el desconocimiento de la existencia de diferentes parámetros que ayudan a un sitio WEB a posicionarse, así como presentar una herramienta web que monitoree el posicionamiento web de la ESPOL.

El análisis SEO que se realiza a un sitio comprende una serie de parámetros que son importantes conocer para alguien que trabaja como Webmaster, sea en una PYME o una corporación, o incluso una institución académica. El presente proyecto está pensado para convertirse en una herramienta que mida cómo nos ven en el Internet, que se ha convertido en una métrica para evaluar cómo estamos y/o cuál es el interés que generamos como institución, individualmente y dentro del dominio de las universidades del país.

1.2. ¿Qué es un buscador?

"Un motor de búsqueda es un sistema informático que indexa archivos almacenados en servidores WEB gracias a su «spider» (o WEB crawler)." [1]

Existen diferentes tipos de buscadores, pero los más usados son: Google, Yahoo!, Live, Ask y AOL. Los buscadores realizan dos procesos básicamente: búsqueda e indexación (construcción/actualización del índice). Para la construcción de los índices los buscadores deben rastrear la WEB en busca de páginas. Para lograr este cometido, los buscadores utilizan un sistema de robots, arañas o rastreadores. Los robots avanzan enlace a enlace por todo el contenido de la red. Este rastreo no es lineal sino que se efectúa sobre las páginas que más cambian o las más relevantes para hacerlo lo más óptimo posible. En este proceso se analiza no sólo las palabras claves o la temática sino varios parámetros que miden la calidad e importancia (criterios de relevancia).

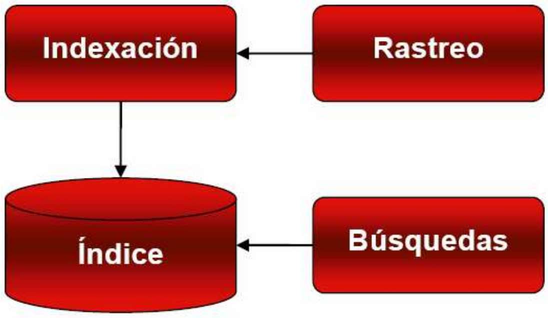

Figura 1. Proceso de búsqueda

Por lo general, los buscadores muestran la información por algún criterio de relevancia, estimando la probabilidad de que una página determinada pueda satisfacer la necesidad del usuario. A modo de resumen, el buscador recibe las palabras a buscar desde la caja de texto del mismo, realiza las consultas en el índice identificando las páginas en donde están presentes dichas palabras. Al mismo tiempo, lee los criterios de relevancia que se crearon en la fase de indexación y las ubica de acuerdo a dichos parámetros de importancia, y los presenta al usuario. Estos criterios de relevancia no son revelados, pero de la experiencia de los usuarios se han detectado dos grandes grupos de incidencia:

- Los intra-site, que se refieren a la información que se puede obtener desde el propio sitio.
- Los extra-site, que son los enlaces externos que apuntan hacia nuestro sitio.

1.3. Barreras de rastreo

La barreras de rastreo son las dificultades técnicas que tienen los robots de los buscadores para posibilitar el rastreo de un sitio WEB, es decir nos imposibilitan estar visibles para los buscadores. De manera general podemos decir que las barreras que los robots de un determinado buscador presentan son:

- Código javascript: Poner cuidado a los enlaces creados por medio de este código, ya que los robots no pueden determinar las rutas formadas de esta manera.
- Flash: No pueden indexar el contenido embebido en estos recursos multimedia.
- Applets de java: Los buscadores no pueden interpretar el código creado en estos componentes java.
- Contenido embebido en imágenes: Al igual que las animaciones en flash y en los applets de java, el contenido visualizado como imagen no puede ser accedido por los robots de los buscadores.
- Formularios: No se puede colocar información en los campos, para obtener nuevas páginas para indexarlas por medio de los robots.

1.4. ¿Qué es el spam en buscadores (spamdexing) ?

Debido al factor económico asociado a la posición de un sitio WEB, sectores de la red concentran su esfuerzo en idear la forma de engañar a los buscadores para hacer creer que una página presenta más relevancia de la que realmente la tiene.
El Spamdexing es uno de los varios métodos de manipular la relevancia de los recursos indexados por un motor de búsqueda.

- Spam de Contenido (Texto Oculto o invisible, Relleno de palabras clave, Relleno de Metatag, Puertas de entrada o pasarela, Sitio de recortes)
- Spam de enlaces (Granja de enlaces, Enlaces ocultos, "Sybil attack" , Wiki spam)

1.5. Prácticas SEO

Para efectos de ubicar o posicionar un sitio WEB, se debe realizar algunas tareas como las siguientes:

- Lograr que otros sitios con temática similar, enlacen con nuestro sitio
- Darse de alta en directorios como Dmoz [2] (http://www.dmoz.org/), y Yahoo! (http://dir.yahoo.com/).
- Construir una amplia base de datos de keywords y frases claves con relación al

negocio. "Google Trends"[3] nos puede ayudar mucho al realizar el análisis sobre la lista de palabras claves sobre las que deseamos realizar SEO, con el objeto de determinar cómo las personas buscan dentro de nuestro modelo de negocio. Una buena alternativa al momento de determinar las palabras claves que irán en el sitio es utilizar la herramienta para palabras clave de google "Keyword Tool External" visitando la dirección https://adwords.google.com.ec/select/KeywordToolExternal.

- Limitar el uso de flash, javascript, frames en la página, ya que esto limita la búsqueda o rastreo del robot en la página al ser visto como espacio plano por el cual no pueden navegar.
- Crear títulos y descripciones pertinentes del contenido de cada página. Estos son como la tarjeta de presentación de la página frente a los buscadores, donde los title y description son puntos iniciales para identificar los términos relevantes.
- Generar un "sitemap" que permita al buscador desplazarse por el sitio de manera ordenada y clara, mejorando su visibilidad.
- Actualizar la página con contenido original de calidad.
- Generar un archivo "robots" de forma correcta. Este archivo indica qué páginas deben indexarse por parte del motor de búsqueda y cuáles bloquearse.

1.6. Robots.txt

Conocido también como "crawler" o "spider" [4], es un archivo de texto plano que debe ser colocado en el directorio raíz como sigue:

- http://dominio/robots.txt

En este archivo se incluye instrucciones para el crawler – programa que inspecciona el WEB – para indicar qué páginas se deben seguir e indexar y a cuáles se deben bloquear el acceso. Esto se hace porque los buscadores pueden indexar el contenido que no nos interesa posicionar o por lo menos no antes de otro contenido (que puede incluso no ser indexado)... Ley de Murphy para el posicionamiento web.

Los buscadores normalmente difieren en los criterios de posicionamiento para las páginas, por lo que aquello que un buscador considera importante para ascender, otro buscador puede considerarlo objeto de descenso.

¿Cómo indicar a Google que deje de indexar ciertas páginas que están optimizadas para otro buscador?... incluir el archivo robots.txt.

User-Agent: (Spider Name)
Disallow: (File Name)

User-Agent corresponde al nombre del spider del buscador mientras que Disallow el nombre del archivo que no deseamos que indexe el spider.

User-Agent: Googlebot (El spider de Google)
Disallow: directorio/pagina-00.html
Disallow: directorio/pagina-01.html
Disallow: directorio/pagina-02.html
Disallow: pagina-03.html

Este código prohíbe el acceso al spider de Google a dos páginas optimizadas para Yahoo! y dos páginas optimizadas para Altavista. De tener permiso Google para acceder a dichas páginas se corre el riesgo de ser penalizados o eliminados de sus búsquedas.

Se puede encontrar una lista más detallada de los diferentes robots, visitando la dirección "http://www.robotstxt.org/db.html".

1.7. Sitemap.xml

El sitemap (mapa de sitio WEB, mapa de sitio o mapa WEB) es un archivo que contiene una lista de las páginas del sitio y que se encuentra accesible para los spiders y los usuarios, organizado de forma jerárquica.

Esto ayuda a los visitantes y a los robots de los motores de búsqueda a encontrar las páginas del sitio. Este archivo ayuda a mejorar el posicionamiento en los buscadores, garantizándonos que las páginas puedan ser encontradas. Es importante sobre todo si se utiliza menú en Flash o Javascript que contienen los enlaces HTML.

Otorgan también una ayuda a la navegación al mostrar una vista general del contenido del sitio WEB. Los sitemaps suelen usar XML, aunque admiten también fuentes WEB RSS y archivos de texto como formato.

Se recomienda poner el archivo sitemap en la raíz[5] como sigue:

- http://dominio/sitemap.xml

2. DISEÑO E IMPLEMENTACIÓN

La adquisición de métricas para evaluar la calidad o cantidad de contenido WEB de las universidades ecuatorianas es el principal problema que presenta nuestra solución. Las métricas a considerar son tomadas de diversos servidores, algunos accedidos mediante APIS provistos por ellos, otras capturando los valores mostrados como contenidos de algún sitio WEB que las provee.

El trabajo explicado más adelante, luego de un análisis donde por importancia se seleccionó determinados indicadores, es llevado a la plataforma WEB por la particularidad de esta de poder ser accedida con recursos mínimos como una conexión a la red y la presencia de un browser en el lado del cliente.

Nuestra solución permite que cada usuario registrado pueda darle seguimiento a su o sus sitios de interés obteniendo información actualizada, en su mayor parte, con retardo de un día; en el peor de los casos, muestra información atrasada con siete días. La mejor manera de explicar el significado de un indicador es mediante gráficos; por ello nuestra aplicación provee al usuario la facilidad de interpretación con estas herramientas. Estos al ser muy heterogéneos hacen complejo el entendimiento al usuario. Por esto diseñamos un dashboard o consola de parámetros donde se explican con semáforos y flechas el estado de cada uno.

2.1. Esquema general de la aplicación

El programa SEO ESPOL es una aplicación WEB, construida para poder ser accedido por usuarios registrados comunes y un usuario administrador. Además se pueden consultar sin necesidad de ser un usuario registrado el estado de spam de cualquier sitio WEB.

Es una aplicación multicapa, desarrollada de esta forma para poder dar mejor mantenimiento y escalabilidad a la misma. Siguiendo el patrón MVC y cumpliendo con estándares WEB para mejor visualización en los diferentes navegadores, intenta ser una herramienta útil de manera que haga fácil el entender la situación de los diferentes sitios WEB registrados.

Revisemos el esquema general de la aplicación:

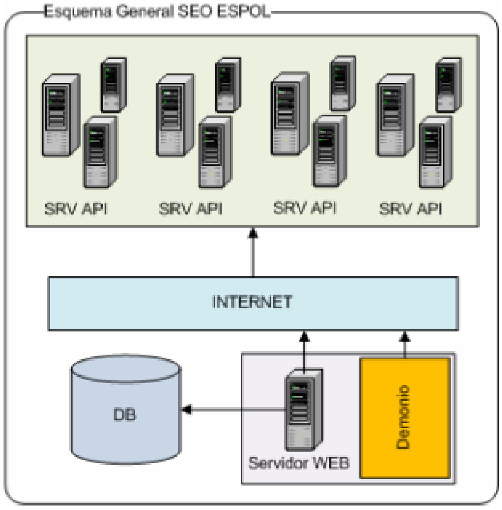

Figura 2. Arquitectura de la aplicación

2.2. Implementación de la aplicación WEB

SEO ESPOL consta de diferentes módulos que describimos a continuación:

Módulo de Ingreso al sistema
Permite acceder a las funcionalidades principales del sistema de acuerdo al rol al que pertenezca el usuario. Existen dos roles:

1. Administrador, que habilita las funciones de control del sistema, permitiendo, dar de alta o baja a usuarios, establecer límites objetivos en las métricas, controlar los valores de búsqueda spam.
2. Usuario, es el módulo general que habilita el seguimiento de los sitios WEB registrados y observados mediante el dashboard. Para acceder al sistema se debe ingresar el usuario y password registrados y validados por el administrador. La figura 3 muestra la pantalla de ingreso al sistema.

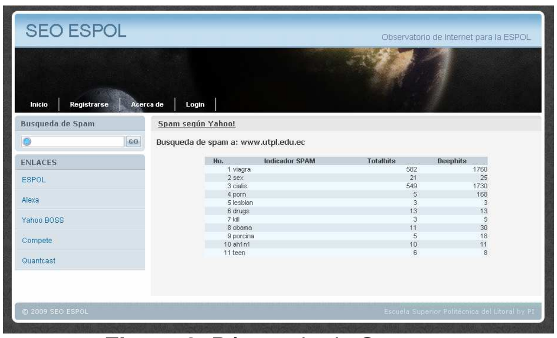

Figura 3. Búsqueda de Spam

Módulo de Presentación de datos
Considerado el corazón de la aplicación, debido a la importancia del mismo de acuerdo a los

objetivos, se encarga de mostrar al usuario la lista de sitios registrados por él, para poder examinar el comportamiento actual del sitio seleccionado de manera fácil y precisa.
Aquí podremos apreciar las diferentes métricas seleccionadas con los valores adquiridos por el módulo de adquisición de datos.

Módulo de Registro de Sitios
El módulo que permite registrar sitios a usuarios previamente registrados y aceptados por el administrador.

Módulo de Filtro Spam
Permite al administrador del sistema ingresar nuevos términos para búsqueda de spam mediante el servicio y criterios de Yahoo!, como también deshabilitar términos que pasaron de moda y por ende no son atractivos para el spam.

Módulo de Configuración de Límites para métricas
Permite al administrador establecer los límites mínimos y máximos de las métricas registradas. De esta manera podemos mantener un control real de los objetivos deseados en el análisis de nuestros sitios.

Módulo Adquisición de datos
Nuestro sistema es en sí una base de datos del comportamiento histórico en el WEB de los diferentes dominios registrados. Ordenados por métricas según grupos de interés, periódicamente se necesita de una actualización de estas métricas para ver y decidir qué camino seguir de manera que pueda optimizarse el sitio WEB.

Para esto se decidió implementar en resumen:

1. Un demonio lanzador de procesos de actualización de métricas,
2. Procesos encargados de consultar las métricas establecidas y guardarlas en la base de datos.
3. Base de datos de métricas para el dashboard.
4. El dashboard o panel que muestra las métricas actualizadas.

Demonio
Un demonio es un programa que se ejecuta en segundo plano sin necesidad de interacción o acción de algún usuario. Es un proceso especial que actúa indefinidamente y esperando la o las condiciones necesarias para su actuar.

Ejemplos comunes de demonios en sistemas Linux son: httpd, sendmail, ftpd, etc.

Nuestro sistema se enfrenta a la necesidad de obtener información de diferentes servicios para los diferentes sitios registrados en la aplicación.
En nuestras primeras pruebas nos dimos cuenta de que la actualización de métricas por sitio WEB se tomaba entre 45 y 300 segundos aproximadamente. Esto depende de la demora en la respuesta del servicio consultado, así como en la complejidad de la respuesta al ser procesada.

Es por esto que se decidió implementar como solución un procesamiento por hilos. En los cuales conociendo que cada petición se procesa independientemente y en paralelo se coloco las llamadas a los API en hilos que son llamados cada cierto tiempo.

Para controlar la llamada a los APIs de manera que no se requiera del comportamiento humano para su efectividad, se decidió establecer un HILO del tipo TIMER. Este hilo tiene como particularidad que se auto ejecuta cada determinado tiempo.

Por las diferencias entre las métricas decidimos separarlas dentro del Timer en dos grupos. Las que se ejecutan a diario y la que se ejecutan cada 7 días. El criterio para establecer esta diferenciación se basa en el costo. El costo de las peticiones tienen dos dimensiones, la dimensión precio/consulta y la dimensión bytes/consulta.

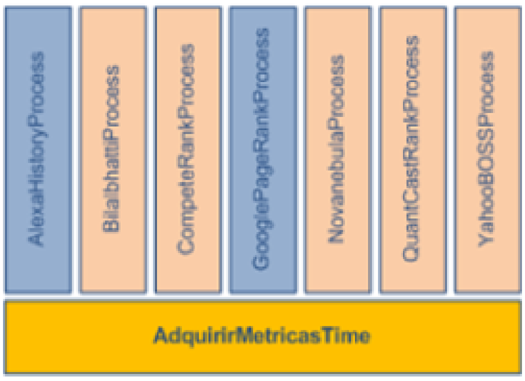

Figura 4: Procesos de adquisición

En el gráfico podemos apreciar al demonio AdquirirMetricasTime que es el encargado de lanzar los diferentes procesos que se encargan por cada sitio registrado en la base de datos, en consultar a los servicios respectivos. Cada proceso a su vez actualiza las métricas establecidas en la base de datos.

AlexaHistoryProcess
Este proceso se ejecuta cada siete días y obtiene la información histórica de tráfico WEB de los

diferentes sitios registrados en la base de datos. Hace uso del API de Alexa el cual se detalla a continuación.

Alexa Web Information Services
Alexa Web Information Services es una plataforma basada en servicios WEB que pretende brindar a los desarrolladores mediante su API información amplia acerca de sus sitios WEB. El acceso a esta información se realiza mediante peticiones de cualquiera de estos dos tipos:

- Query request.- Son peticiones HTTP GET con respuestas XML
- SOAP request.- Son peticiones cuyo paso de parámetros son encapsulados en formato SOAP. La arquitectura del sistema maneja únicamente Query request.

Ejemplo:
http://awis.amazonaws.com/?AWSAccessKeyId=98765432123451238&Timestamp=2007-01-26T01%3A15%3A38.000Z&Signature=oQkiPZUtQ9PlTI2l4OTRA8fjYsM%3D&Version=2005-07-11 &Action=UrlInfo&ResponseGroup=Rank&Url=espol.edu.ec

En nuestro ejemplo podemos observar diferentes parámetros formando la url, que es pasada al servlet para ser procesada. Estos parámetros son:

· **AWSAccessKeyId**.- Representa el AlexaID de un usuario registrado en AWIS. En dicho registro debe el usuario definir un método de pago. Esta definición se realiza agregando a su cuenta la información de una o más tarjetas de crédito, de donde mensualmente se debitará los valores correspondientes a las consultas realizadas.

El paquete de 1000 consultas tiene un costo de 0.15USD, sin embargo el usuario del servicio solo pagará la parte proporcional a su consumo. Este AlexaID por los motivos expuestos debe manejarse con cuidado.

· **Timestamp**.- Se refiere al tiempo actual en formato UTC. Es tomada en consideración por AWIS para responder o no a la petición, ya que si difiere de 15 minutos en relación a los servidores de Amazon la respuesta se denegará. Este parámetro pasa codificado.

· **Action**.- Indica lo que se va a pedir en la URL. En general define lo que se puede pedir. Los valores que puede tomar son:

· **CategoryBrowse**: Devuelve la lista de sitios ordenados por categorías por traffic rank.

· **CategoryListing**: Parecido a CategoryBrowse pero basado en contenido.

· **SitesLinkingIn**: Devuelve una lista de los sitios WEB que enlazan a el sitio WEB observado. Este dato es actualizado cada dos meses.

· **UrlInfo**: Obtiene información acerca del sitio WEB analizado tal como es: que tan popular es, sitios relacionados, información de contacto.

· **TrafficHistory**: Esta acción retorna el ranking diario de Alexa, límites por millones de usuarios y páginas vistas. Los datos que almacena AWIS pueden ser consultados siempre que comiencen no antes de Junio del 2007.

· **Signature**.- Es la firma de la URL. Es calculada con la concatenación del parámetro Action con el Timestamp en conformidad con el estándar RFC-2104 HMAC-SHA1, para encriptar esta cadena, usando como clave el Secret Access Key provisto por Amazon.

· **Versión**.- Permitirá según Amazon al API seguir consultando incluso cuando la versión del mismo cambie. Actualmente es la 2005-07-11 y también es un parámetro opcional.

· **ResponseGroup**.- Varía de acuerdo al Action y se puede requerir varios separados con coma. Permite filtrar los datos del Action y es obligatorio de acuerdo al Action.

· **URL**.- Se refiere al sitio a consultar. Una vez ejecutada la petición, AWIS devuelve un XML con la respuesta el cual se lo parsea para extraer los datos que interesan y almacenarlo.

Nuestro proceso que es lanzado por el demonio cada siete días, se encarga de consultar el ranking histórico de los últimos días faltantes disponibles. Es decir consulta para cada sitio los datos desde el último día guardado más uno hasta la última fecha que provea el servicio.

Generalmente debe ser 7 días.

BilalbhattiProcess
Este proceso es lanzado cada 24 horas por el demonio para consultar en el sitio http://bilalbhatti.com/ las siguientes métricas para cada sitio registrado en la base de datos:

· Google Pagerank
· Google backlink

· Google indexed
· Alexa popularity o rank
· Alexa backlink
· AllTheWeb result
· Altavista result
· Technorati rank
· Technorati inblogs

Se estableció este sitio como base de esta información por las pruebas que se realizaron que demostraron la fidelidad de los datos. Además de la necesidad imperiosa de minimizar el tiempo de adquisición de métricas por sitio.

Para procesar este requerimiento debemos enviar una solicitud HTTP GET mediante la siguiente URL base:

http://bilalbhatti.com/checksite.php?extra=yes&url=[sitio]

Donde [sitio] es la dirección de la que se requiere estas métricas.

La respuesta del sitio es en formato HTML, y ha sido minimizada al código estrictamente necesario para mostrar los valores de las métricas.

CompeteRankProcess
Compete es un sitio que ofrece un API para la adquisición libre de su ranking que es basado en el parámetro estadístico de mediciones web llamado Unique Visitors o Reach, que considera solamente una vez al usuario por determinado tiempo, incluso si este ha ingresado varias veces al sitio web, generalmente es un mes.

Para consultar a este API se requirió de un registro en el sitio, donde se nos otorgo acceso mediante una clave dada y que se requiere suministrarla cada vez que el llamado al API es realizado.

Este llamado es mediante una solicitud HTTP GET y requiere de una URL de la siguiente forma:

http://api.compete.com/fastcgi/MI?d=[sitio]&ver=3&apikey=vdny39yptsbukrcm3wcfsjhy&size=large

Donde [sitio], el dominio a ser observado en este servicio.

La respuesta del servicio es mediante XML.

GooglePageRankProcess
Este proceso lanzado por el demonio cada siete días. El pagerank como ya hemos mencionado no se actualiza a diario sino entre dos y tres meses, por lo cual no necesitamos lanzar este proceso diariamente, sin embargo por la no difusión de la última vez que se actualizo este parámetro por parte de Google, hemos decidido que se haga en periodos de siete días.

Este servicio lo consumimos del sitio http://icons.geek-tools.org/, mediante una petición HTTP GET y con respuesta de texto plano. La url de llamado al API es:

http://icons.geek-tools.org/api/show_pr.php?url=[sitio]

NovanebulaProcess
Este proceso es lanzado cada día para consultar en Novanebula que es un sitio que basado en diferentes parámetros SEO calcula un precio para el sitio web referido. Para acceder a este API debemos hacer una petición HTTP GET mediante la siguiente url:

http://www.novanebula.net/money/?site=[sitio]

Donde sitio es el dominio al que queremos estimar su valor.

La respuesta del API es dada en formato XML.

QuantCastRankProcess
Quantcast es un sitio que se dedica a realizar análisis web. Para adquirir el valor del ranking según quantcast.com el demonio lanza este proceso cada día mediante una petición HTTP GET cuya url es:

http://www.quantcast.com/[sitio]

Donde sitio es el dominio que analizaremos mediante esta herramienta. El valor de respuesta es HTML, el cual es procesado mediante este proceso.

YahooBOSSProcess
Este proceso lanzado cada 24 horas, hace uso del API de Yahoo! Llamado BOSS (Build your own search services). Este proyecto es una iniciativa de Yahoo! para que terceros puedan hacer uso de su infraestructura y logren crear productos de búsqueda haciendo uso de sus datos. Hacemos uso de este API para obtener:

· Cantidad de spam aproximado por sitio web.
· Páginas indexadas en Yahoo!.
· Páginas que enlazan a nuestro sitio.

El llamado a este servicio implica enviar una solicitud HTTP GET mediante la siguiente plantilla de url:

http://boss.yahooapis.com/ysearch/se_inlink/v1/ [sitio]?appid={yourBOSSappid}&format=xml &count=1

Donde [sitio] es la dirección que vamos a revisar, appid es el ID provisto por el servicio una vez que hemos procedido debidamente con el registro, format indica el formato que debe usar el servidor para devolver la respuesta y count indica el índice del primer dato devuelto.

Nosotros hemos decidido que la respuesta sea recibida en XML, para mantener la consistencia con los demás servicios.

Una vez ejecutados los procesos, estos se encargan de almacenar en la base de datos por sitio, métrica y fecha el valor obtenido en cada caso. Cabe indicar que esta toma de datos al hacerse en segundo plano no interfiere en ningún momento con el manejo del sistema, pero si podría no mostrar los datos actualizados hasta que los hilos terminen su ejecución.

AskProcess
Mediante este hilo podemos traer a nuestra base de datos valores desde los servidores de ASK, sitio muy reconocido por contener buenos índices de contenido del web.

Este API no brinda a la aplicación dos importantes métricas como son:

· **Ask index**: Que se refiere a las páginas que están indexadas en el servidor de ASK.

· **Ask backlink**: Que se refiere a las páginas de terceros que hacen referencia al sitio consultado por nosotros. Para poder acceder a este servicio vamos a realizar un Query Request de tipo HTTP GET, donde para cada una de las métricas arriba señaladas tenemos un formato de URL a indicar:

http://es.ask.com/web?q=site:[url1]

Donde [url1] es la dirección del sitio que estamos monitoreando la cantidad de contenido indexado.

http://es.ask.com/web?q=inlink:[url2]

Donde [url2] es la dirección del sitio que estamos monitoreando la cantidad de sitios que lo refieren.

BingProcess
El API del buscador Bing, requiere de un registro para la obtención de un API ID que se enviará cada vez que se requiera hacer una petición de información.

Por ser un buscador joven no presenta las mismas ventajas ni madurez de los API de Yahoo!. Sin embargo logramos obtener de él, el número de páginas indexadas por su robot.

Es por esto que para ver como se encuentra indexado el contenido publicado por las universidades del país, hacemos uso de este API también que tiene la siguiente URL para realizar el QUERY REQUEST de tipo HTTP GET.

http://api.search.live.net/xml.aspx?Appid=4A8 D5344BDC9A7A86819CE8046FA6827E3D43 D7A&sources=web+image&query=[url3]

Donde Appid es el identificador provisto por live.net para realizar las consultas sobre los datos de los datacenters de Bing.

También [url3] define la dirección que estamos necesitando obtener la cantidad de contenido publicado y accesible en internet y que ha sido indexado por Bing.

2.3. Diseño de la base de datos

Hemos diseñado la base de datos del sistema que permite almacenar:

· Los usuarios del sistema
· Las métricas por sitio
· El ranking de Alexa
· El control de spam

Podemos revisar su estructura en la figura 9:

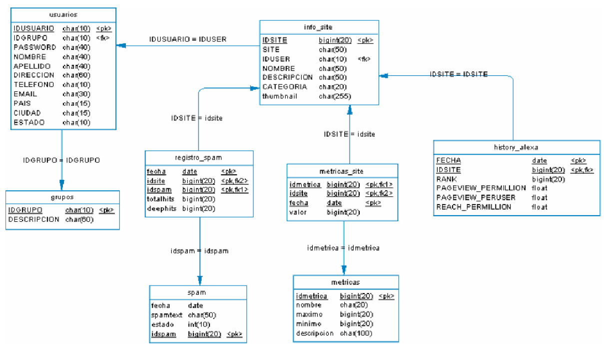

Figura 5: Diseño físico de la base de datos

3. Conclusiones

Toda información que esté en el WEB, debe aspirar a ser accesible y visible a los usuarios, principalmente porque estamos en un mundo tan competitivo. Los buscadores son los que canalizan el tráfico que se genera en la red y que día a día crece en volumen. De allí la importancia de aparecer en los primeros lugares dentro de las búsquedas de los internautas.

Se han desarrollado un conjunto de herramientas SEO (Search Engine Optimizer) que servirán de apoyo al WEB máster de la ESPOL en el proceso de optimización, del website de la universidad, en los motores de búsqueda. De esta manera, se implementa una solución que representa una ventaja competitiva frente a sitios similares, al poder evaluar cómo estamos en popularidad frente a sitios similares que representan nuestra competencia para, a partir de este conocimiento, incorporar las acciones necesarias que nos permitan seguir a la vanguardia de las universidades del país.

El hecho de tener un observatorio es siempre poder mirar más allá, y de esta observación obtener una ventaja en el entorno. Pretendemos brindar a las universidades las métricas suficientes para que vean como pueden crecer y hacer que la competencia por el bien de la educación nacional siempre este presente. Pero la competencia debe ser leal siempre. Existen lastimosamente prácticas malsanas que provocan que existan valores medidos con incorrecta información, pues de una u otra manera, se ha permitido que el spam WEB afecte a estos parámetros, cosa que hemos detectado y que intentamos, con el sistema, ayudar a medir.

4. Referencias

[1] WIKIPEDIA. 2009. Motor de Búsqueda. (Disponible en: http://es.wikipedia.org/wiki/Motor_de_busqueda. Consultado el: 12 de octubre de 2009).

[2] MÁSTER ONLINE EN BUSCADORES. Selección de unidades didácticas 2008. (Disponible en: http://www.masterenbuscadores.com/mbus_2007_2008.pdf. Consultado el: 1 de noviembre de 2009).

[3] http://google.com/trends.

[4] ROBOTSTXT. (Disponible en: http://www.robotstxt.org/. Consultado el: 30 de octubre de 2009).

[5] SITEMAPS. Preguntas frecuentes. ¿Dónde puedo colocar mi Sitemap? (Disponible en: http://www.sitemaps.org/es/faq.php#faq_sitemap_location. Consultado el: 3 de noviembre de 2009).