Internet Bot para la Obtención y Análisis de Información Empleando el Motor de Búsqueda de Google

Cevallos D.*; Cevallos F.*; Bernal I.*; Mejía D.*

**Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica, Quito, Ecuador*
e-mail: (davidcepn@yahoo.com;epnfernando@gmail.com; {ivan.bernal; david.mejia}@epn.edu.ec)

Resumen: Este artículo presenta la implementación de un *Internet bot* que puede ser utilizado por una o varias aplicaciones cliente a través de un servicio web. El *Internet bot*, empleando el motor de búsqueda de Google y un listado de palabras clave y URL de sitios web ingresados por el usuario, realiza varias búsquedas para obtener las páginas web de los sitios indicados que contengan información relacionada con las palabras clave. El *Internet bot* analiza la información de cada página web encontrada y, mediante un algoritmo de puntuaciones propio del *Internet bot*, determina por cada página web únicamente la información que con mayor probabilidad contiene el dato exacto que el usuario busca. De esta manera, el usuario puede encontrar de manera sencilla y rápida información precisa en Internet y no tan sólo los URL de las páginas web que contienen dicha información.

Palabras clave: Internet bot, servicio web, Google, Uniform Resource Locator, página web, sitio web, texto plano, algoritmo.

Abstract: This paper presents the implementation of an *Internet bot* that can be used by one or more client applications through a web service. The *Internet bot, by* using the Google search engine, a list of keywords and URL of websites specified by users, performs several searches for getting the web pages of the indicated sites that may have information related with the provided keywords. The *Internet bot* analyzes the information of each web page found during the Google search and by using a scores algorithm, specifically developed for the *Internet bot*, determines, for each web page, only the information that most likely contains the precise data that the user is looking for. Thus, the user can easily and quickly find accurate information in the Internet and not just the URL for the pages that contain such information as is tha case with a Google search.

Keywords: Internet bot, web service, Google, Uniform Resource Locator, web page, website, plain text, algorithm.

1. INTRODUCCIÓN

En la actualidad se puede encontrar un sinfín de información en la web, por lo que el uso de buscadores adquiere cada día mayor importancia. El motor de búsqueda de Google es uno de los principales buscadores gracias a su algoritmo *PageRank* y, sin lugar a dudas, el más utilizado a nivel mundial [3].

Sin embargo, el uso de un buscador por sí solo no garantiza que la información resultante de una búsqueda se acerque al dato preciso que el usuario pretende encontrar ni que la información corresponda a un sitio web de confianza con información fiable; por lo que en este artículo se presenta la implementación de un *Internet bot*, ofrecido a través de un servicio web, que hace uso del motor de búsqueda de Google para obtener los URL (*Uniform Resource Locator*) de páginas web contenidas en determinados sitios web de confianza que el usuario indica y, a partir de dichas páginas, analiza la información que cada una contiene para presentar únicamente información de importancia para el usuario.

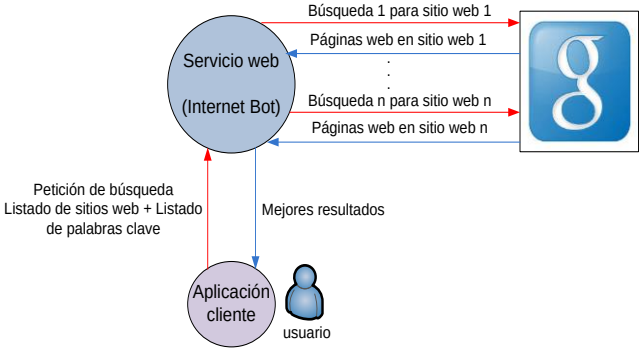

Figura 1. Funcionamiento del sistema

La Fig. 1 presenta el funcionamiento del sistema. El usuario realiza a través de la aplicación cliente una petición de búsqueda al servicio web, indicando para ello un listado de sitios web de confianza en donde se debe realizar la búsqueda y un listado de palabras clave.

Artículo recibido el 14 de diciembre de 2014. Este trabajo fue financiado por la Escuela Politécnica Nacional (proyecto semilla PIS-12-30) y con el soporte del MINTEL y la RedAUTI en la que participa la EPN, parte de las "Iniciativas sobre TV Digital" que se desarrollan en el DETRI. Información de contacto: Iván M. Bernal, e-mail: ivan.bernal@epn.edu.ec, Tel: 593-2-2507-144 ext.2338. Escuela Politécnica Nacional. Ladrón de Guevara E11 - 253. Quito, Ecuador.

Por cada sitio web indicado, el *Internet bot* realiza una búsqueda en Google utilizando el listado de palabras clave ingresado por el usuario. La búsqueda que se realiza es la misma que realizaría el usuario directamente en el buscador, ingresando cada palabra clave separada por un espacio en blanco y colocando, al final del listado, el URL del sitio web.
Por cada búsqueda realizada se obtiene, a partir de los diez primeros resultados retornados por Google, un listado con los URL correspondientes a las páginas web contenidas en el sitio, las cuales podrían tener información útil para el usuario.
Finalmente, el *Internet bot* convierte la información de cada página web encontrada por Google en texto plano y se la separa en párrafos. Cada párrafo de una página web es puntuado mediante un algoritmo de puntuaciones que se desarrolló específicamente para el *Internet bot*, de manera que se presente al usuario, como resultado de la búsqueda, únicamente el párrafo con el puntaje más alto de cada página web.
El servicio web y la aplicación cliente fueron implementados mediante la plataforma Visual Studio .NET 2012 con el lenguaje de programación C#. El servicio web emplea la tecnología WCF (*Windows Communication Foundation*) y el protocolo SOAP (*Simple Object Access Protocol*) para la comunicación con la aplicación cliente.
El artículo está organizado de la siguiente manera: En la sección 2 se presenta el mecanismo que emplea el *Internet bot* para obtener la información de Internet empleando el motor de búsqueda de Google, y en la sección 3 el mecanismo empleado para obtener la información de una página web en texto plano. En la sección 4 se presenta el algoritmo de puntuaciones del *Internet bot* con el cual se analiza la información en texto plano de cada página web y finalmente se determina aquella información de interés para el usuario. En la sección 5 se presenta la aplicación cliente a través de la cual el usuario puede hacer uso del *Internet bot*. En la sección 6 se presentan algunos ejemplos de la funcionalidad del *Internet bot*, y en la sección 7 se realiza un análisis de las ventajas y limitaciones del *Internet bot* implementado, así como también una comparación con otras tecnologías existentes. Finalmente, en la sección 8 se presentan las conclusiones del artículo.

2. BÚSQUEDA DE INFORMACIÓN

2.1 *Búsqueda empleando el motor de búsqueda de Google*

Con el fin de realizar una búsqueda en Google, el *Internet bot* emplea la clase `WebClient` de .NET, disponible a través del espacio de nombres `System.Net` [8]. Un objeto de esta clase permite descargar la información de un recurso web a través del método `DownloadData()` cuyo argumento es el URL del recurso [8]. En este caso, este URL corresponde a una petición de búsqueda a Google, la cual tiene la siguiente estructura
`http://www.google.com.ec/search?q=búsqueda`. La palabra `búsqueda` consiste en una cadena de texto conformada por el listado de palabras clave con las cuales se realiza la búsqueda y el URL del sitio web, separados por el símbolo +. Al emplear este URL, Google retornará los diez primeros resultados de la búsqueda. El método `DownloadData()` retorna un *array* de *bytes*, correspondiente al resultado de la búsqueda en Google, el cual puede ser convertido en una cadena de texto empleando la clase abstracta `Encoding`, disponible a través del espacio de nombres `System.Text` [6]. El resultado final es una cadena de texto en formato HTML (*HyperText Markup Language*) en la cual están contenidos los URL de las páginas web resultantes de la búsqueda, que se pueden obtener haciendo uso de la biblioteca `HTMLAgilityPack`.
`HTMLAgilityPack` es una biblioteca de clases que permite analizar el código HTML de páginas web de manera sencilla para poder obtener la información que se desee; este proceso es también denominado *screen scraping*. La biblioteca fue desarrollada para C# y puede obtenerse de [2]. La Tabla 1 presenta algunas de las clases de la biblioteca, que permiten manipular el código de páginas web sin tener que preocuparse de si el código HTML está bien formado o no, ya que encontrar páginas web cuyo código HTML contenga errores es bastante común en Internet.

Tabla 1. Algunas clases de la biblioteca `HTMLAgilitypack`

Clase	Descripción
`HtmlDocument`	Modela un documento con código HTML.
`HtmlNode`	Representa un nodo HTML, el cual puede ser hijo de otro nodo y a la vez contener nodos hijos.
`HtmlAttribute`	Representa una propiedad de la etiqueta HTML de un nodo.
`HtmlWebException`	Permite manejar excepciones producidas por código HTML mal formado.
`HtmlDocument`	Modela un documento con código HTML.
`HtmlNode`	Representa un nodo HTML, el cual puede ser hijo de otro nodo y a la vez contener nodos hijos.

Una de las principales características de `HTMLAgilityPack` es que permite obtener, mediante consultas `XPATH`[2], una colección de nodos HTML correspondientes a determinadas etiquetas HTML que cumplan con la condición de la consulta; también se puede obtener de manera sencilla el valor de determinadas propiedades de las etiquetas HTML de dichos nodos. Para ello, se debe proveer como argumento al método `SelectNodes()` de un objeto de la clase `HtmlNode` una ruta `XPATH` como una cadena de texto. Particularmente, el nodo padre de un documento HTML se puede obtener mediante la propiedad `DocumentNode` del objeto de la clase `HtmlDocument` que representa el código HTML de la página web.
Dado que en un resultado de búsqueda retornado por Google los enlaces hacia las páginas web están contenidos en la propiedad `href` de las etiquetas con el *tag name* `a`, mediante la consulta `XPATH` de estructura `//a[@href]` se podrá obtener fácilmente una colección con los nodos HTML cuyo *tag name* sea `a` y contengan la propiedad `href`. De esta manera, se logra obtener una colección con todos los URL de las páginas web correspondientes a la búsqueda en Google.

[2] `XPATH` (ruta XML) evalúa una expresión de tipo `//tag_name[@propiedad]` para obtener un subconjunto del conjunto de nodos HTML que cumplan con la condición de la expresión [9].

2.2 *Obtención de los URL de páginas web*

Por defecto, Google emplea un mecanismo propietario de codificación de URL, por lo que los URL obtenidos en la búsqueda no pueden ser empleados directamente para acceder a las páginas web [3]; un análisis detallado del mecanismo de codificación de URL que Google realiza excede el alcance de esta investigación.

Es suficiente mencionar que, de manera general, el URL codificado está conformado por varios campos separados por el símbolo `&` y que el URL sin codificar se puede obtener a través de la manipulación del campo inicial, es decir, de la cadena de texto antes de la aparición del primer símbolo `&`. El proceso de decodificación del URL retornado por Google se detalla en la Fig. 2. En este proceso se ha añadido un paso de selección, en el que únicamente se toman los URL correspondientes a páginas web con contenido HTML, para su posterior análisis.

Si la página web está contenida dentro del sitio web indicado previamente por el usuario, el valor de la propiedad `href` empezará con la secuencia `/url?q=` seguido del URL del sitio web. Si no es así, la página web corresponde a otro sitio web por lo que se descarta. Ciertos caracteres especiales dentro del URL retornado por Google están codificados.

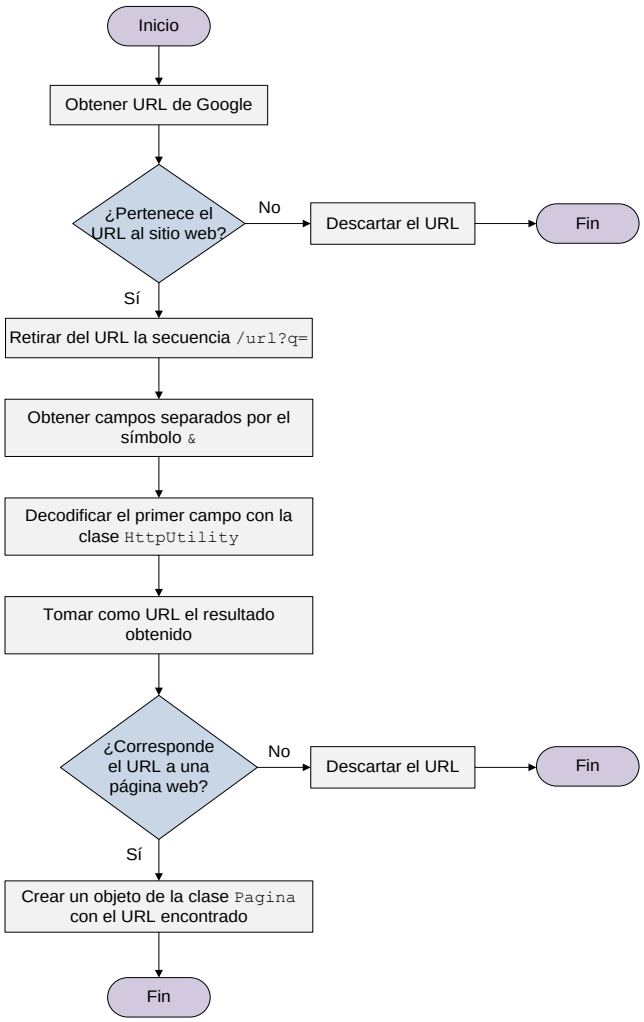

Figura 2. Proceso de decodificación y selección de un URL obtenido mediante una búsqueda en Google

Por ejemplo, los espacios en blanco son reemplazados con un signo `+`, y algunos caracteres especiales son reemplazados por un signo `%` seguido por su equivalente hexadecimal en código ASCII [10]. .NET ofrece la clase `HttpUtility` disponible en el espacio de nombres `System.Web` [7]. El método `UrlDecode()` de esta clase recibe como argumento una cadena de texto que posee caracteres especiales codificados y retorna como resultado la cadena de texto decodificada.

El URL original (sin la codificación de Google) puede ser recuperado si se elimina de la cadena de texto del URL codificado por Google la secuencia inicial `/url?q=` y todo el texto restante después del primer `&` (incluyendo al símbolo `&`); y se pasa la cadena resultante al método `UrlDecode()` de la clase `HttpUtility`.

Finalmente, como paso adicional, se analiza la extensión del recurso al cual corresponde el URL obtenido con el fin de asegurarse de que el URL corresponda a una página web con contenido HTML y no a recursos como, por ejemplo, documentos o aplicaciones en línea. Por lo tanto, si el URL culmina en extensiones tales como .indd, .pdf, .ppt, .doc, .xls, .pptx, .docx, .xlsx, .exe, entre otras, dicho URL es descartado. Si el URL es válido, entonces se crea un objeto de la clase `Pagina`[3] (el cual representa una página web) con el URL encontrado.

3. OBTENCIÓN DE LA INFORMACIÓN DE UNA PÁGINA WEB EN TEXTO PLANO

Una vez que el *Internet bot* ha obtenido los URL correspondientes a las distintas páginas web en las que puede existir información de interés para el usuario, debe obtenerse el código HTML de cada página web mediante un objeto de la clase `WebClient` y el empleo de la clase `Encoding`, de forma similar al mecanismo de obtención del resultado de búsqueda en Google.

La información de la página web está contenida en nodos HTML por lo que, para facilitar su posterior análisis mediante el algoritmo de puntuaciones, es necesario obtener dicha información en texto plano, eliminando las etiquetas HTML y tomando únicamente la información de interés; es decir, tomando como una cadena de texto únicamente la información correspondiente a la que el usuario puede leer al abrir la página web.

Esta tarea se la puede realizar mediante la biblioteca `HTMLAgilityPack` y la implementación de un método recursivo que recorra cada uno de los nodos HTML, e ir añadiendo la información encontrada en cada nodo a una cadena de texto inicialmente vacía. La Fig. 3 describe el método recursivo que implementa el *Internet bot* para este propósito.

Antes de invocar al método por primera vez, se obtiene el nodo padre del documento HTML mediante la propiedad `DocumentNode` del objeto de la clase `HtmlDocument` correspondiente al código HTML de la página web.

[3] La clase `Pagina` y la clase `Respuesta`, que se mencionan en la próxima sección, son clases que se implementaron para el *Internet bot* con el objetivo de representar y gestionar a una página web y a una línea/párrafo de texto respectivamente.

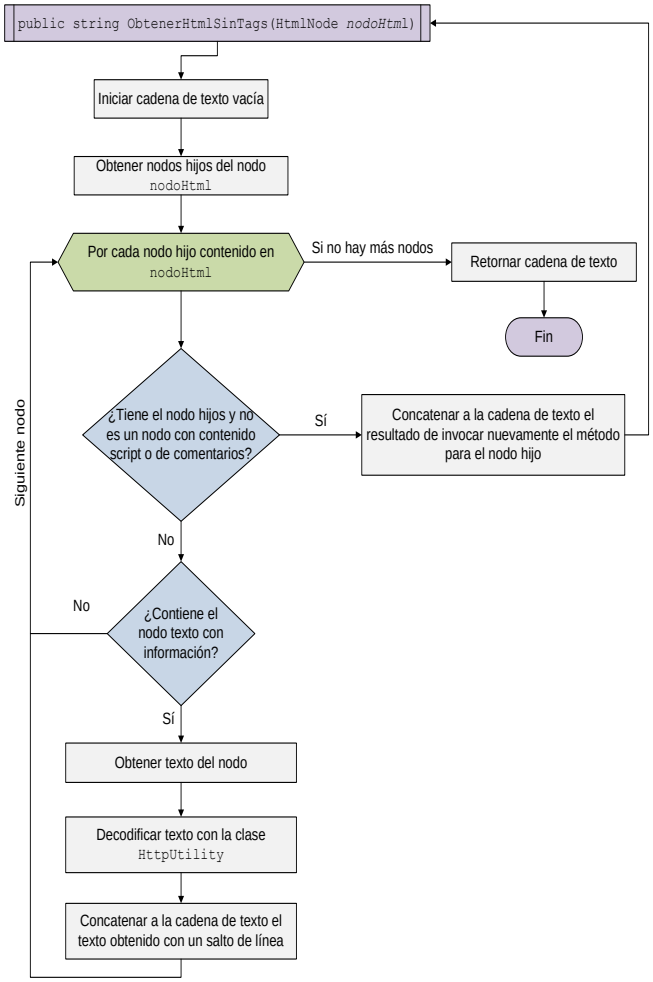

Figura 3 Método recursivo para obtención de la información de una página web en texto plano

Luego, al invocarse el método recursivo, se pasa el nodo padre como argumento.
El nodo padre, al igual que el resto de nodos del documento HTML, es manipulado mediante la creación de un objeto de la clase `HtmlNode`. La Tabla 2 detalla los atributos de la clase `HtmlNode` empleados en el método recursivo para manipular los nodos. Si el nodo no corresponde a un nodo con una etiqueta *script* o de comentario y posee nodos hijos, el método se invoca nuevamente con el nodo como argumento. Si el nodo no posee más nodos hijos y corresponde a una etiqueta que contiene información, se añade a la cadena el texto contenido en el nodo HTML, el cual se puede obtener mediante el atributo `InnerText` del objeto de la clase `HtmlNode` correspondiente al nodo.
Al culminar el método recursivo, el resultado será la información contenida en la página web como una cadena de texto.

Tabla 2 Algunos atributos de la clase `HtmlNode` de la biblioteca `HTMLAgilitypack`

Atributo	Descripción
`ChildNodes`	Colección de objetos `HtmlNode` correspondientes a los nodos hijos del nodo.
`OriginalName`	Nombre original de la etiqueta HTML del nodo que permite reconocer el contenido de la etiqueta. Por ejemplo: • **`script:`** La etiqueta contiene código de *scripting*. • **`#comment:`** La etiqueta contiene un comentario. • **`#text:`** La etiqueta contiene información.
`InnerText`	Texto contenido dentro de la etiqueta HTML.

4. ALGORITMO DE PUNTUACIONES

El algoritmo de puntuaciones del *Internet bot*, el cual se describe en la Fig. 4, es efectuado para cada página web correspondiente a los URL obtenidos con la búsqueda en Google. En primer lugar se debe obtener la información contenida en la página web como texto plano, mediante el proceso descrito en la sección anterior, para luego proceder a puntuar cada línea.
El algoritmo emplea el listado de palabras clave ingresado por el usuario para puntuar a cada línea. Por cada palabra clave se analiza cada línea para determinar si ésta contiene la palabra, para lo cual se emplean tres criterios:

- Si la línea contiene en alguna parte de su texto la palabra clave, se suman dos puntos.
- Si la línea contiene exactamente la palabra clave más otro texto adicional, se suman tres puntos.
- Si la línea contiene exactamente la palabra clave sin otro texto adicional, se suman cuatro puntos.

La puntuación total de cada línea corresponde a la suma de los puntuajes obtenidos por su análisis con cada palabra clave. Para obtener resultados más precisos, el algoritmo añade un punto adicional a cada línea que sea inmediatamente anterior o posterior a una línea con un puntaje mayor o igual a dos, pues es probable que el dato exacto que busca el usuario se encuentre entre estas líneas. Si la línea ha recibido puntaje (es decir, tiene un puntaje mayor o igual a uno) se crea un objeto de la clase `Respuesta`, que representa la línea. Finalmente, aquellas líneas que no han recibido puntaje son eliminadas.
Luego, el texto resultante es separado en párrafos. La división en párrafos que realiza el algoritmo no corresponde a párrafos gramaticales, sino a un conjunto de líneas que recopilan la mayor cantidad de coincidencias con las palabras clave, y en las que es más probable se encuentre el dato exacto que el usuario está buscando. Las líneas con puntaje igual a uno determinan el inicio y final de cada párrafo. La puntuación total del parráfo se obtiene entonces con la suma de los puntajes de cada una de las líneas que lo conforman. Por cada párrafo se crea un objeto de la clase `Respuesta` que representa al párrafo.
El párrafo con el puntaje más alto es el que se presenta al usuario como resultado de búsqueda en la página web, pues existe una mayor probabilidad de que en dicho párrafo se encuentre la información que el usuario está buscando.
Puede darse el caso de que ninguna línea haya recibido puntaje de acuerdo a los criterios del algoritmo, de manera que todas las líneas hayan sido eliminadas. En este caso, el algoritmo considera que en la página web no existe información de interés para el usuario y no retornará resultados para dicha página.

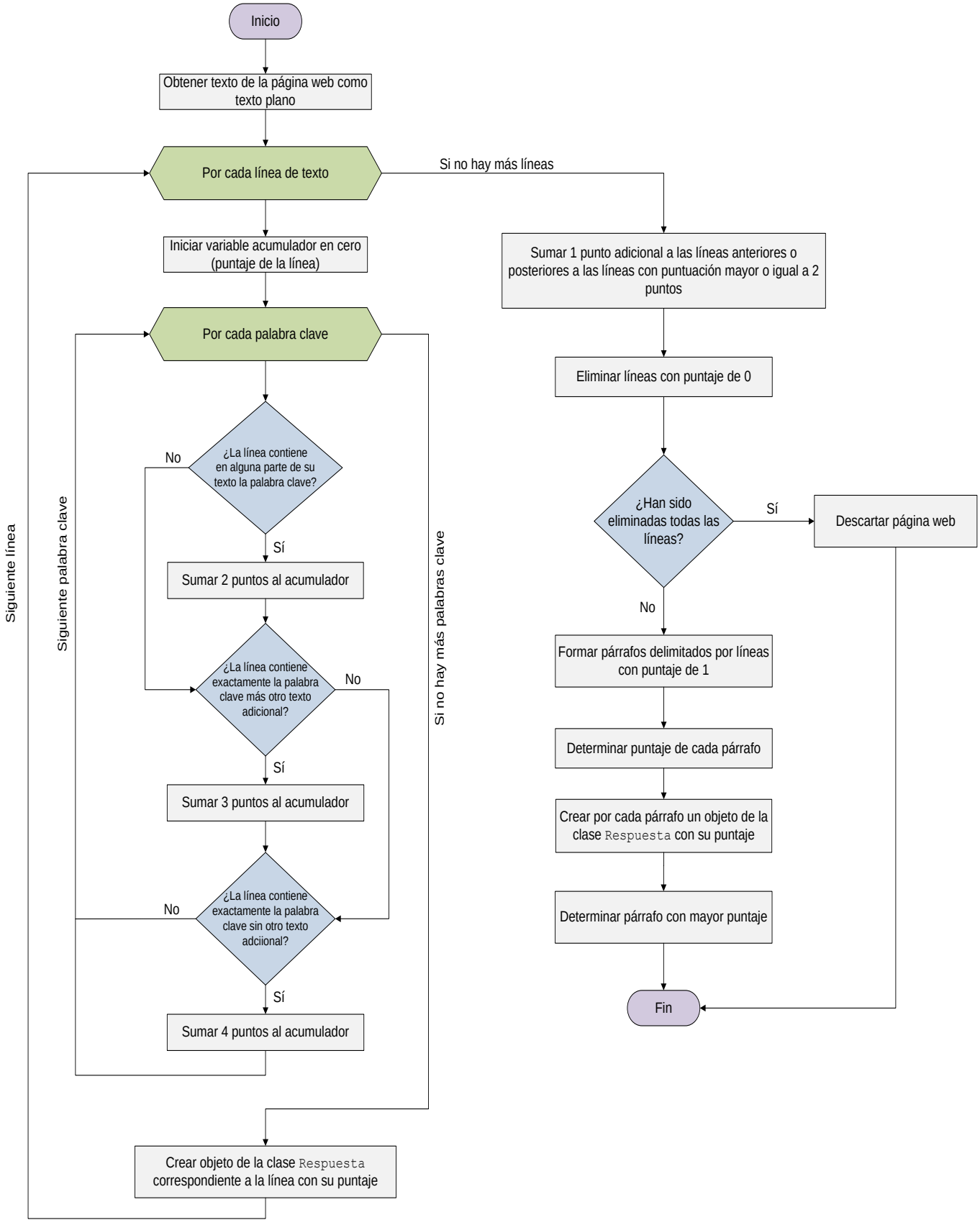

Figura 4. Algoritmo de puntuaciones

5. APLICACIÓN CLIENTE

5.1 Interfaz gráfica

La aplicación cliente brinda una interfaz gráfica al usuario de manera que pueda hacer uso del servicio web que ofrece la funcionalidad del *Internet bot*.

La Fig. 5 presenta la ventana de trabajo que la aplicación cliente brinda para que el usuario pueda ingresar el listado general de los URL correspondientes a los sitios web de confianza que se emplearán para las distintas búsquedas de datos que desee realizar. Al abrir esta ventana, el listado de sitios web de confianza ingresado se presentará al usuario. El usuario puede agregar más sitios web así como eliminarlos.

La ventana principal de trabajo de la aplicación se presenta en la Fig. 6. Esta ventana provee la interfaz gráfica a través de la cual el usuario puede crear, cambiar o eliminar temas correspondientes a un tópico general; así como crear, cambiar o eliminar subtemas (subtópicos correspondientes al tópico general). Esto permite al usuario realizar búsquedas correspondientes a varias temáticas de forma organizada.

Para cada subtema, la aplicación permite crear una matriz de las dimensiones que el usuario desee. Cada celda de la matriz podrá almacenar información. La ventana principal de trabajo dispone de un navegador interactivo a través del cual el usuario puede movilizarse entre las distintas celdas de la matriz creada de un subtema seleccionado. El usuario puede ingresar el contenido de la celda de forma manual o a través del *Internet bot* que brinda el servicio web.

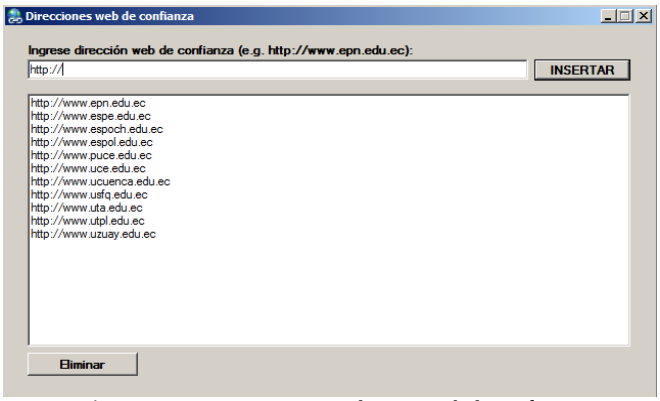

Figura 5. Ventana para ingreso de sitios web de confianza

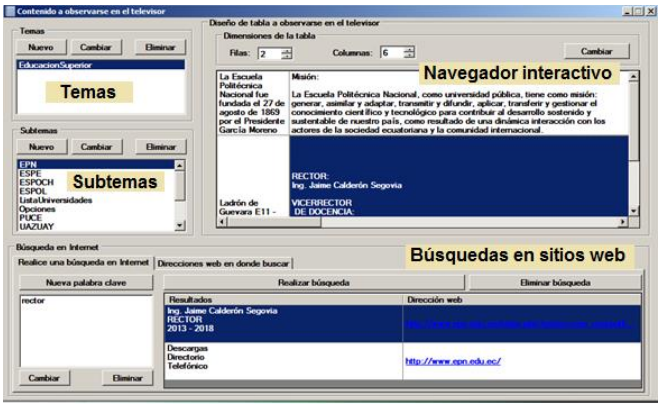

Figura 6. Ventana principal de trabajo

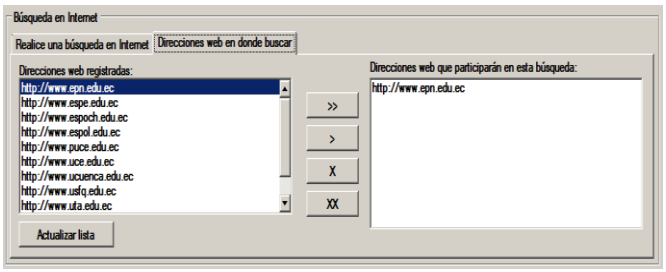

Figura 7. Pestaña *"Direcciones web donde buscar"*

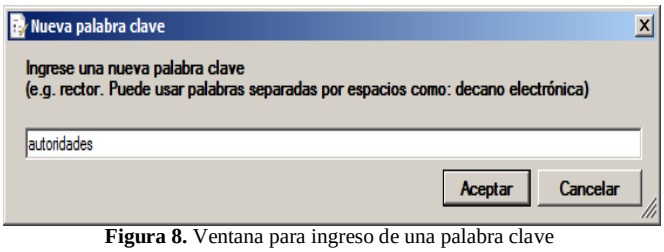

Figura 8. Ventana para ingreso de una palabra clave

En la parte inferior de la ventana principal de trabajo, el usuario cuenta con los recursos necesarios para poder realizar la búsqueda de un dato específico para la celda seleccionada con el navegador interactivo. El sitio o los sitios web seleccionados para realizar la búsqueda, con el fin de obtener la información a almacenarse en la celda, pueden ser seleccionados de entre las direcciones de confianza en la pestaña *"Direcciones web en donde buscar"* que se presenta en la Fig. 7.

Con el fin de realizar la búsqueda de un dato específico con los sitios web indicados, el usuario deberá ingresar un listado de palabras clave haciendo un clic sobre el botón "Nueva palabra clave". Al hacer clic sobre este botón, se presentará al usuario una ventana como la de la Fig. 8 para que ingrese la nueva palabra clave.

El listado de palabras clave ingresado se presentará en la parte inferior de la ventana principal de la aplicación. La búsqueda se realizará al hacer clic sobre el botón *"Realizar búsqueda"*. La aplicación lista el mejor resultado obtenido (párrafo mejor puntuado) de cada página web encontrada por el *Internet bot* con la ayuda de Google y presenta junto al resultado el URL de dicha página. Conforme el usuario se moviliza entre las distintas celdas con el navegador interactivo, los resultados de las búsquedas realizadas para cada celda se desplegarán en la parte inferior de la ventana principal de trabajo.

Adicionalmente a todas estas funcionalidades, la aplicación cliente permite especificar horarios de búsqueda para la realización de búsquedas automáticas.

5.2 Configuración de horarios de búsqueda automática de información

Con el fin de que el usuario pueda disponer de información actualizada, empleando la aplicación cliente se puede hacer uso de la opción de configuración de horarios de búsqueda automática de información que brinda el *Internet bot*.

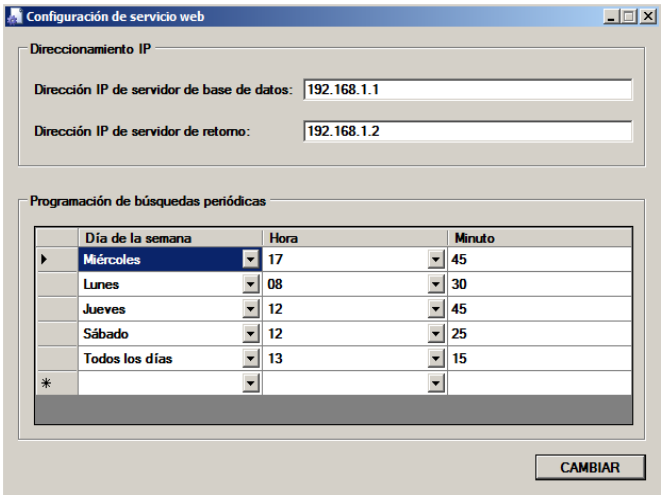

Figura 9. Ventana de configuración de horarios de búsqueda automática de información

La Fig. 9 presenta la ventana de configuración de horarios de búsqueda automática de información. Un horario de búsqueda está determinado por el día de la semana, la hora y los minutos en los cuales se desea que el servicio web ejecute automáticamente la funcionalidad del *Internet bot*. El usuario puede especificar uno o varios horarios.

Cada vez que se ejecuta una búsqueda automática, la aplicación cliente recibe la información encontrada por el *Internet bot* y la presenta en la parte inferior de la ventana como sugerencias para el usuario; de esta manera, el usuario puede determinar si la información encontrada es de utilidad o no.

Por otro lado, la ventana de configuración de horarios permite, opcionalmente, establecer la dirección de un servidor de bases de datos y de un servidor de retorno. Si se especifica una dirección de un servidor de bases de datos, el usuario puede emplear la aplicación cliente para enviar la información al servidor de bases de datos externo correspondiente a la dirección especificada. De igual manera, si se especifica una dirección de un servidor de retorno, el usuario puede enviar la información a un servidor externo (como por ejemplo un servidor web, ftp, entre otros) en donde se almacenará la información para cualquier fin o utilidad posterior que desee el usuario. Esta funcionalidad de almacenamiento es independiente de si la información fue obtenida como resultado de la ejecución de una búsqueda automática (mediante el establecimiento de un horario) o de una búsqueda manual (haciendo clic sobre el botón *"Realizar búsqueda"*).

6. EJEMPLO DE FUNCIONALIDAD

A manera de ejemplo, haciendo uso de la aplicación cliente, se han realizado varias búsquedas para encontrar, mediante el *Internet bot*, información de interés general respecto a universidades del Ecuador.

La Fig. 12 presenta el resultado obtenido al realizar la búsqueda para obtener los nombres de las autoridades de la Pontificia Universidad Católica de Quito (PUCE). Las palabras clave ingresadas para esta búsqueda son:

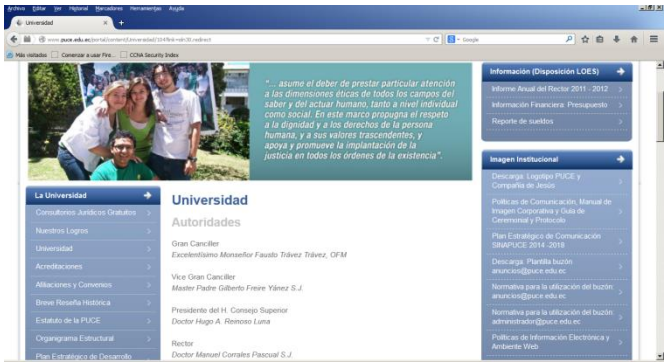

Figura 10. Página web de la cual el *Internet bot* obtuvo información de las autoridades de la PUCE

`autoridades`, `rector` y `vicerrector`; y el único sitio web indicado es `http://www.puce.edu.ec`, correspondiente al sitio web de la universidad. Al hacer un clic derecho sobre el URL de la página web y seleccionar la opción *"Abrir"* del menú contextual emergente, se presentará al usuario en el navegador la página web de la cual el *Internet bot* obtuvo la información, la cual se presenta en la Fig. 10.

La Fig. 13 presenta la búsqueda realizada para encontrar la misión de la Escuela Politécnica Nacional (EPN). La única palabra clave ingresada para esta búsqueda es `misión`; y el único sitio web indicado es `http://www.epn.edu.ec` correspondiente al sitio web de la universidad. La Fig. 11 presenta la página web de la cual el *Internet bot* obtuvo la información.

Como se puede apreciar en los ejemplos, la información que se presenta en el resultado es bastante precisa al dato que se pretende buscar. Mediante la configuración de horarios de búsqueda automática se puede obtener información permanentemente actualizada, lo cual puede ser de interés para el usuario dependiendo de sus necesidades.

Como ya se mencionó, la información obtenida a través del *Internet bot* puede ser almacenada en un servidor de bases de datos con el fin de ser empleada posteriormente. A continuación se presenta un ejemplo de esta funcionalidad aplicada en el campo de la interactividad en televisión digital. La Fig. 14 presenta el diagrama general de este ejemplo.

Una vez que la información de las universidades es obtenida mediante el *Internet bot*, el usuario (empleando la aplicación cliente) almacena dicha información en un servidor de bases de datos.

Figura 11. Página web de la cual el *Internet bot* obtuvo información de la misión de la EPN

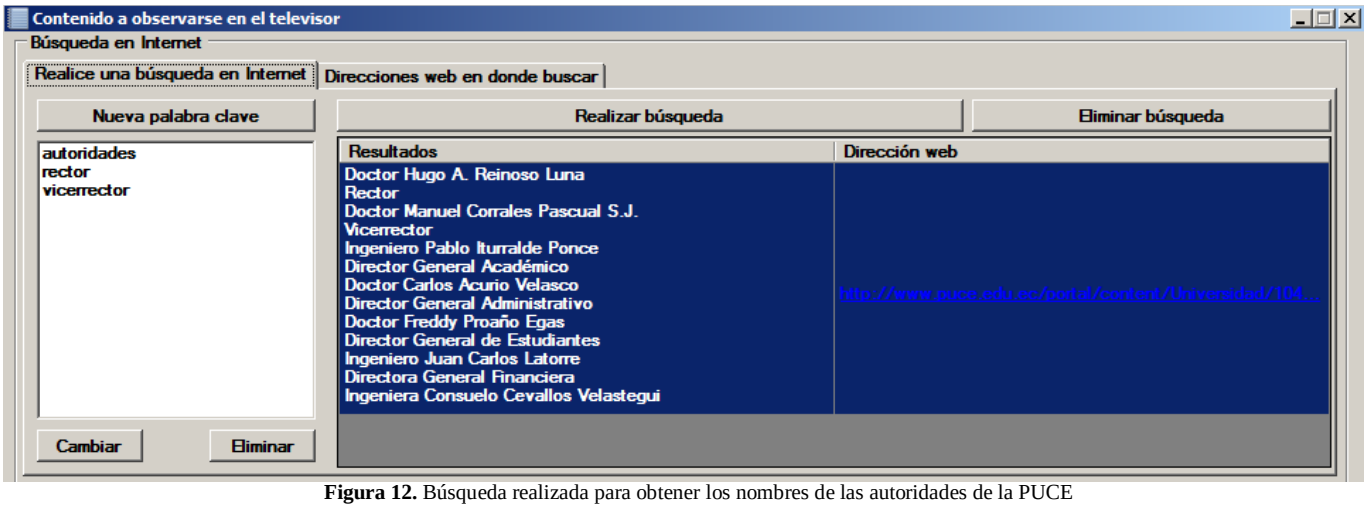

Figura 12. Búsqueda realizada para obtener los nombres de las autoridades de la PUCE

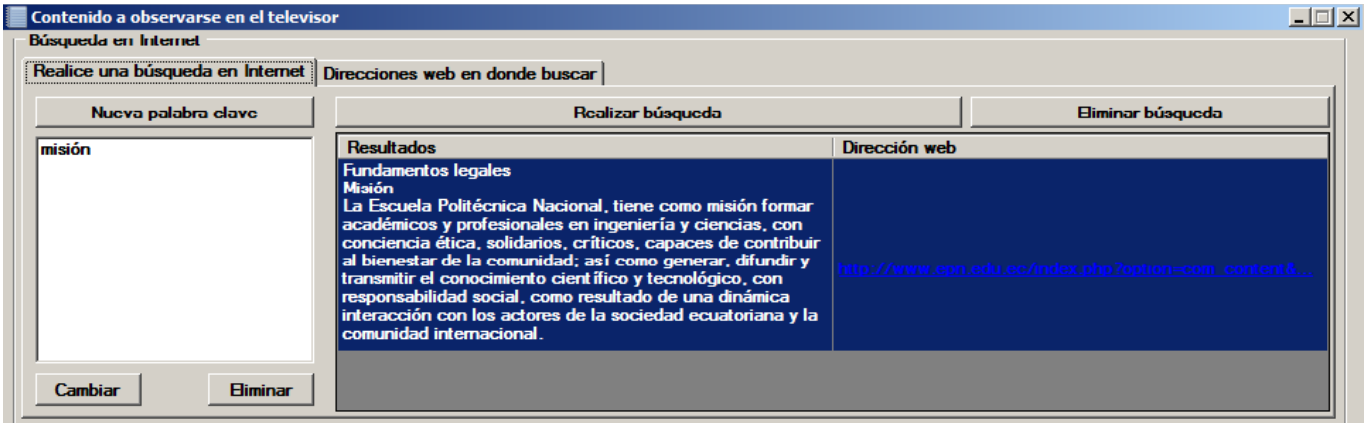

Figura 13. Búsqueda realizada para obtener la misión de la EPN

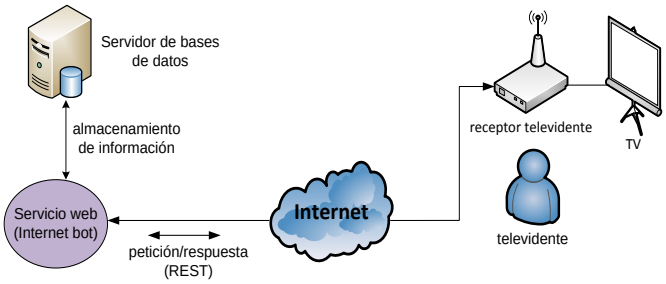

Figura 14. Diagrama general de ejemplo de almacenamiento de información obtenida mediante el *Internet bot*

El servicio web está conectado a través de Internet con el receptor del televidente en donde se ejecuta una aplicación interactiva de televisión digital. El servicio web emplea la tecnología REST para transmitir la información de las universidades (obtenida mediante el *Internet bot* y almacenada en el servidor de bases de datos) cada vez que la aplicación interactiva solicita dicha información. De esta manera, la aplicación interactiva podrá presentar al televidente en la pantalla del televisor la información obtenida mediante el *Internet bot*. Si además se emplea la funcionalidad de búsquedas automáticas de información, cada vez que se encuentre nueva información se almacenará en el servidor de bases de datos. De esta manera, el televidente podrá contar con información actualizada.

7. VENTAJAS, LIMITACIONES Y COMPARACIÓN DEL INTERNET BOT CON OTRAS TECNOLOGÍAS EXISTENTES

En esta sección se analizan algunas de las ventajas y limitaciones del *Internet bot* implementado, y se realiza una comparación del mismo con otras tecnologías ya existentes.

7.1 Ventajas

El *Internet bot* tiene varias ventajas en la obtención y análisis de información en Internet. La principal ventaja (y la cual comprende el estado del arte de esta investigación) radica en que el *Internet bot* desarrollado permite obtener información de determinados sitios web en un tiempo mucho menor que el que tomaría explorar cada sitio recursivamente en búsqueda de dicha información. A este aspecto se suma la ventaja de que la información que el *Internet bot* obtiene comprende datos que, con mayor probabilidad, corresponden a la información precisa que el usuario está buscando y que

pueden contestar a sus interrogantes gracias al algoritmo de puntuaciones propio del *Internet bot*.

Otra ventaja del *Internet bot* implementado es la flexibilidad que éste brinda. Así por ejemplo, el *Internet bot* permite determinar horarios de búsqueda de información para realizar búsquedas automáticas, en lugar de tan solo permitir la realización de búsquedas manuales (es decir, búsquedas que el usuario indica realizar al hacer clic sobre un botón). Aunque el *Internet bot* ha sido programado para obtener los diez primeros resultados de la búsqueda de Google (pues en los diez primeros resultados retornados por Google probablemente se encuentra la información de mayor interés para el usuario), dicho parámetro puede ser fácilmente modificable para que en lugar de analizar únicamente los diez primeros resultados se pueda analizar la cantidad de resultados que se desee.

La idea original con la cual se implementó el *Internet bot* implica que el usuario ingrese los sitios web de su confianza con el fin de obtener información fiable, ya que encontrar sitios web con información no fiable en Internet es algo bastante común. Sin embargo, la ventaja del *Internet bot* implementado radica en que el usuario puede prescindir (si así lo desea) de especificar los sitios web para realizar las búsquedas. En ese caso, el *Internet bot* realizará las búsquedas retornando información proveniente de cualquier sitio web, de acuerdo a los resultados de la búsqueda en Google.

Por otro lado, si bien el *Internet bot* implementado hace uso del motor de búsqueda de Google, debe notarse que si se cambia en el código del *Internet bot* el URL del buscador y si se omite el proceso de decodificación de los URL, el *Internet bot* puede ser capaz de obtener información empleando otros buscadores como Yahoo, Bing, Ask, Gigablast, entre otros. Cabe mencionar que este mismo mecanismo también podría ser empleado para obtener y analizar información retornada no por tan solo buscadores sino también, por ejemplo, redes sociales como Twitter o Facebook, lo cual puede ser una mejora del *bot* que le permitiría obtener información de varias fuentes (no de tan sólo buscadores). Finalmente, como ya se expuso en la sección anterior, el *Internet bot* brinda una importante ventaja al permitir, a través de la aplicación cliente, que la información que éste retorna pueda ser almacenada en un servidor de bases de datos y/o algún servidor externo.

7.2 Limitaciones

Existen ciertas funcionalidades que el *Internet bot* no considera en su implementación y que podrían ser posibles mejoras. En primer lugar, el algoritmo de puntuaciones considera únicamente la frecuencia de aparición de las palabras clave en la información y el cómo son presentadas. No se toma en cuenta casos de polisemia (una palabra con varios significados), homonimia (palabras de misma escritura pero con significados distintos) ni el contexto en el cual aparece la palabra. De igual manera, no se analizan aspectos como, por ejemplo, el tipo de letra de la palabra, el tamaño, el color o si la palabra está marcada en negrita o no, lo cual podría ayudar a discriminar si la información es relevante o no. Así, si la palabra está marcada en negrita o si es parte de un título, es más probable que la información del contexto sea de mayor importancia que aquel cuando la misma palabra aparece sin negrita o es parte de un párrafo.

A esto se suma el hecho de que el *Internet bot* no es capaz de determinar por sí mismo si la información de un sitio web es actualizada o no, y debe confiar en este aspecto en el motor de búsqueda de Google. El *Internet bot* no analiza la fecha de las páginas web y confía en que Google retornará los resultados más actualizados en cada búsqueda.

Finalmente, otra limitante del *Internet bot* radica en el tema de seguridad. El *Internet bot* no discrimina si la información obtenida es proveniente de un sitio web seguro que posea un certificado digital o al cual se pueda acceder mediante una sesión HTTPS (*HyperText Transfer Protocol Secure*), lo cual puede ser un aspecto de mucha importancia en ciertos casos.

Aunque todas las observaciones expuestas son actuales limitantes, podrían ser implementadas a futuro con el fin de mejorar la funcionalidad del *Internet bot*.

7.3 Comparación con otras tecnologías existentes

En la actualidad se impulsa la investigación de buscadores semánticos. Un buscador semántico se basa en el uso de la semántica web y, a diferencia de un buscador común, efectúa la búsqueda tomando en consideración el significado de las palabras con el fin de que la búsqueda sea más precisa sin basarse en simples etiquetas [4]. En la actualidad existen varios buscadores semánticos como por ejemplo Kngine, Hakia, Lexxe, Kosmix, DuckDuckGo, Evri, Powerset, entre muchos otros [5].

La diferencia entre el *Internet bot* y un buscador semántico radica en dos aspectos básicos:

- Un buscador semántico busca información con precisión empleando semántica web, mientras que el *Internet bot* busca información con precisión analizando la información obtenida como resultado de búsquedas en Google.
- Un buscador semántico toma en cuenta el significado de las palabras y su contexto, mientras que el *Internet bot* se basa únicamente en la frecuencia de aparición de las palabras y el cómo son presentadas (algoritmo de puntuaciones).

Sin embargo, cabe mencionar que la semántica web es aún un campo en investigación y varios buscadores semánticos están siendo empleados aún en versiones beta [4]. El objetivo del *Internet bot* implementado es lograr los mismos resultados que al emplear un buscador semántico pero sin hacer uso de la semántica web (aún en desarrollo). Vale rescatar el hecho de que el *Internet bot* podría ser empleado con un buscador semántico en lugar de Google si así se desea. La Fig. 15 presenta, como ejemplo, la búsqueda realizada en el buscador semántico DuckDuckGo para obtener la misión de la EPN, la cual se obtuvo anteriormente empleando el *Internet bot*. Nótese que se ha realizado la búsqueda en este buscador

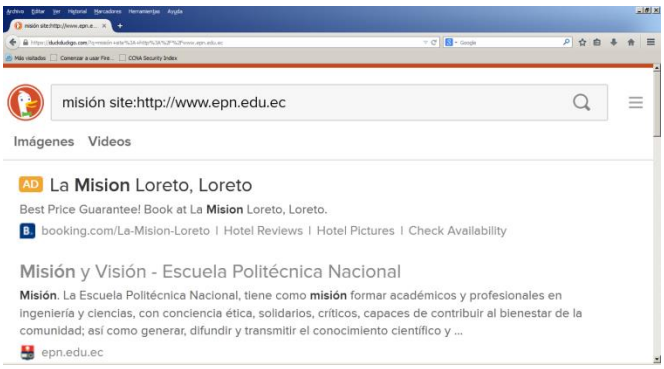

Figura 15. Búsqueda de información con el buscador semántico DuckDuckGo

señalando el sitio web en donde buscar a través de la palabra `site`.

Finalmente, vale la pena hacer una comparación entre el *Internet bot* implementado y un *web crawler*. Un *web crawler* es un *bot* de búsqueda que metódica y automáticamente navega sitios web en Internet con el fin de crear un índice (*web indexing*) de los sitios web que navega, el cual pueda ser usado posteriormente por un motor de búsqueda [11]. Como ejemplos de *web crawlers* tenemos a GoogleBot (usado por Google), Yahoo Slurp (usado por Yahoo) y MSNBot (usado por Bing) [1].

La principal diferencia entre el *Internet bot* implementado y un *web crawler* radica en que el *Internet bot* no busca información recursivamente en cada sitio web por sí mismo, sino que hace uso del motor de búsqueda de Google para esta tarea; por lo tanto el *Internet bot* es mucho más rápido. De igual manera, el *Internet bot* no crea un índice de los sitios web que visita.

8. CONCLUSIONES

El *Internet bot* implementado facilita la obtención de información precisa en sitios web al presentar los mejores resultados obtenidos mediante un algoritmo de puntuaciones. Para su funcionamiento se hizo uso del motor de búsqueda de Google, el cual permitió obtener los URL de las páginas web contenidas dentro de los sitios web indicados por el usuario en un tiempo mucho menor que al explorar recursivamente cada sitio.

El motor de búsqueda de Google emplea, a diferencia de otros buscadores, un mecanismo de codificación de URL; por lo tanto no es posible hacer uso directamente de estos URL sin antes someterlos a un proceso de decodificación. El servicio de búsqueda puede trabajar con otros buscadores si se omite la ejecución del proceso de decodificación de URL.

El desarrollo de *bots* de búsqueda capaces de obtener información que responda con exactitud a las preguntas del navegante en Internet es aún un campo en desarrollo.

REFERENCIAS

[1] AHFX. GoogleBot and other spiders. [Online]. Disponible en: http://www.ahfx.net/weblog/39

[2] CodePlex. HTMLAgilityPack. [Online]. Disponible en: http://htmlagilitypack.codeplex.com

[3] G. Colouris, J. Dollimore, T. Kindberg y G. Blair, "Distributed Systems Concepts and Design", 5ta ed., Addison-Wesley, Ed. USA: Pearson, 2012.

[4] Google. Buscadores semánticos. [Online]. Disponible en: https://sites.google.com/site/buscadoressemanticos/-que-es-un-buscador-semantico

[5] Google. Ejemplos de buscadores semánticos. [Online]. Disponible en: https://sites.google.com/site/buscadoressemanticos/ejemplos-de-buscadores-semanticos

[6] Microsoft. Encoding (Clase). [Online]. Disponible en: http://msdn.microsoft.com/es-es/library/system.text.encoding%28v=vs.110%29.aspx

[7] Microsoft. HttpUtility (Clase). [Online]. Disponible en: http://msdn.microsoft.com/es-es/library/system.web.httputility%28v=vs.110%29.aspx

[8] Microsoft. WebClient (Clase). [Online]. Disponible en: http://msdn.microsoft.com/es es/library/system.net.webclient%28v=vs.110%29.aspx

[9] Visual Studio. Sintaxis de XPATH. [Online]. Disponible en: http://msdn.microsoft.com/es-es/library/vstudio/ms256471%28v=vs.100%29.aspx

[10] Wikinsonpc. Codificando y decodificando una dirección URL. [Online]. Disponible en: http://www.wilkinsonpc.com.co/free/articulos/codificar-decodificar-url.html

[11] Wikipedia. WebCrawler. [Online]. Disponible en: http://en.wikipedia.org/wiki/Web_crawler