UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
CONSEJO DE ESTUDIOS DE POSTGRADO
POSTGRADO EN COMPUTACIÓN

ALEJANDRÍA INTELIGENTE

Un experimento Web semántico

Trabajo de grado presentado como
requisito final para optar al grado de
MAGISTER SCIENTIAE en Computación

Ícaro Alzuru C.
Tutores:
Jacinto Dávila
José G. Silva T.

Mérida – Venezuela
Febrero 2007

RESUMEN

Este documento presenta el resultado de un experimento con tecnologías Web semánticas implementadas sobre un sistema de gestión de información conocido como Alejandría.

Se contextualiza qué es la Web semántica, se muestran las tecnologías semánticas y Web semánticas que se tomaron en consideración, se presenta y justifica el uso de Alejandría en el experimento y se propone una arquitectura de utilización de algunas tecnologías semánticas en las aplicaciones de manejo de información en general (usando Alejandría como caso de prueba). Como resultado se logra una mejora en la recuperación de información obtenida a partir de las búsquedas que realizan los usuarios a través de la interfaz de la aplicación.

Se utilizan: OWL para la catalogación, estandarización y estructuración de la información compartida; WordNet para la interrelación de descriptores (mejora de búsquedas); y gramáticas en Prolog para el reconocimiento de preguntas realizadas en lenguaje natural. Estas tecnologías fueron combinadas y aplicadas en el sistema de gestión de información; para mostrar en forma conceptual y a través de ejemplos típicos, dentro de un dominio de conocimientos específico, que la arquitectura propuesta funciona satisfactoriamente.

CONTENIDO

INTRODUCCIÓN

El presente escrito documenta un experimento dirigido a incorporar las tecnologías de Web Semántica (así denominadas) a un sistema de gestión de información llamado Alejandría. El lector encontrará en las siguientes secciones una explicación general de las tecnologías Web semánticas, una breve introducción al sistema de gestión de información utilizado (Alejandría) y los detalles de la extensión hacia la semántica propuesta y desarrollada para el sistema.

El World Wide Web Consortium (W3C), órgano rector de la World Wide Web, ha dedicado esfuerzos a la creación y promoción de varios estándares para la **Web Semántica [14]**.

De forma tímida, algunos sitios Web y aplicaciones de escritorio están siendo construidos utilizando estándares y tecnologías Web semánticas. Se puede presagiar que se verá una aparición gradual de productos basados en este tipo de tecnologías y es probable que en algún momento lleguen a convertirse en una práctica habitual en la Internet: Si esto sucede, los usuarios de todo el mundo dispondrán de una red más colaborativa y orientada a los significados, antes que a los símbolos, haciendo realidad el sueño de Tim-Berners Lee (Creador del World Wide Web).

El problema que aborda la Web semántica compete tanto a las aplicaciones Web como a aplicaciones tradicionales de manejo de información. No se trata sólo de catalogar información, pues estándares como **MARC [9]** tienen ya un camino recorrido y un tiempo considerable en el mercado. Si bien se desea catalogar la información de una forma sencilla y estándar, las tecnologías Web semánticas buscan permitir al computador establecer relaciones y realizar inferencias acerca de entidades (significados) representados en la información compartida. Se trata de explotar el potencial de cómputo con que se cuenta en la actualidad y obtener mejores sistemas de manejo de información. Se persigue ir más allá del computador como una (sofisticada) máquina de escribir que permite almacenar y recuperar textos no interpretados: Hacia un sistema automático en donde el texto escrito pueda ser "entendido" por el software, para ayudar a los humanos en el procesamiento de la información.

Este procesamiento avanzado de la información no es sencillo de lograr. Si se desea que el computador "reconozca" que ciertas palabras dentro de un texto corresponden, por ejemplo, al nombre del autor; se le debe indicar esto de forma explícita usando, por ejemplo, **Metadatos**[1]. El formato de especificación de estos metadatos debe ser tan general, o estándar, como sea posible,

[1] Metadatos: Datos estructurados que describen las características de un recurso de información **[22]**.

para que otras aplicaciones o personas sean capaces de entenderlos. Es aquí donde las especificaciones del W3C (**XML**, **RDF**, **RDFS**, **OWL**, etc.) proporcionan herramientas prácticas y donde iniciativas como **Dublin Core [6]**, para la especificación de distintos tipos de metadatos, transversales entre dominios de conocimientos, son realmente útiles.

Los actuales buscadores en Internet, para poder dar respuesta a las consultas que realizan los usuarios, se ayudan de métodos estadísticos (de aparición de palabras y/o de referencias dentro de Internet) **[7]**. Estos métodos han sido bastante útiles hasta la fecha y logran obtener resultados considerablemente buenos **[32]**. Pero, ¿podríamos mejorar estos resultados con la utilización de la tecnología Web semántica? ¿Valdrá la pena el trabajo de catalogación que involucra la introducción de metadatos en los textos y el establecimiento de relaciones semánticas entre los metadatos usados en un cierto dominio de conocimiento?

El presente trabajo de grado, realiza un experimento Web semántico con una aplicación de gestión de información (**Alejandría**). Se pretende probar la utilidad de estas herramientas semánticas y dar respuesta a las interrogantes planteadas en el párrafo anterior.

En el capítulo 1 se explica qué es la Web Semántica, qué es Alejandría y los problemas que afrontan en la actualidad las aplicaciones de manejo de información para realizar búsquedas efectivas.

En el capítulo 2 se plantea el problema general de la semántica en las búsquedas de información a través de sistemas informáticos.

En el capítulo 3 se proponen un conjunto de tecnologías para abordar el problema del manejo semántico de la información y de la mejora de las búsquedas en los sistemas informáticos.

En el capítulo 4 se presenta cómo se implementaron en Alejandría las tecnologías semánticas propuestas en el capítulo 3 para abordar en forma concreta el problema general antes mencionado.

1.- La Web Semántica y los sistemas de gestión de información

1.1.- ¿Qué es la Web Semántica?

La Web Semántica fue ideada por Tim Berners-Lee, creador del World Wide Web, como una mejora o extensión de la Web actual, donde los datos serán "definidos y enlazados en una forma en que puedan ser utilizados por máquinas, no sólo con propósitos de mostrarlos, sino para automatización, integración y reutilización de los datos a través de varias aplicaciones" **[21]**.

La Web Semántica involucra la definición de varios estándares y la construcción y puesta en funcionamiento de tecnologías que permiten a las computadoras realizar un manejo de la información enfocado en su significado.

La Web Semántica no implica enseñar a las computadoras a procesar un texto e interpretar, deducir o razonar para que sepa de qué trata y cómo se relaciona con otros objetos de su entorno. Es algo mucho más sencillo que eso: Es la catalogación de los textos, sus partes y los objetos allí representados, de forma estándar, estableciendo relaciones entre objetos y metadatos y entre metadatos, de forma que, posteriormente, resulte posible realizar mejores búsquedas y se puedan emplear técnicas de Inteligencia Artificial para la consecución de tareas en el ámbito de la información.

1.2.- Problemas en las búsquedas por Internet y en los sistemas de gestión de información

La mayoría de las aplicaciones informáticas que permiten realizar búsquedas sobre campos de texto, no poseen ningún mecanismo especial para la mejora de la calidad de las consultas; simplemente utilizan consultas a tablas de bases de datos con el comando like '%palabra%' para determinar los registros que hablan sobre cierta palabra o tema.

Por otro lado, los motores de búsqueda de Internet sí utilizan algunas técnicas más elaboradas para la mejora de la calidad del resultado en las consultas que realizan los usuarios. Por ejemplo Google, utiliza métodos de coincidencia textual y se ayuda con un sistema de cálculo de la relevancia de las páginas conocido como PageRank **[7]**. El PageRank consiste en darle mayor relevancia (puntos) a las páginas que son más referenciadas por otras páginas, además, valen más los puntos de las páginas que a su vez, tienen mayor cantidad de referencias. Internamente, Google utiliza enormes mapas de hiperenlaces para el cálculo del PageRank. Esto permite presentarle al usuario una lista ordenada por relevancia, basada en su contenido y en su PageRank. Además del PageRank, Google utiliza técnicas de coincidencia textual como **[32]**:

- Dar mayor importancia a las páginas donde los términos que se están buscando se encuentran próximos.
- Llevar un registro del tamaño de las letras y de si están en negritas, para dar mayor relevancia a los documentos donde las palabras buscadas aparecen en lo que parecieran ser títulos.

En la actualidad, las técnicas de búsqueda se basan en la sintaxis de las palabras. En algunas aplicaciones de escritorio, se consiguen todavía casos como que: Al buscar la palabra "autos" (porque se desea saber sobre automóviles en general), el sistema retorna todos los textos que contienen el vocablo "autos": Incluyendo aquellos textos que contienen la palabra "cautos"; reduciendo la calidad del resultado.

En la mayoría de las ocasiones, se desea que al buscar información sobre un concepto; por ejemplo, "autos", el sistema traiga los documentos que tratan sobre automóviles. Con lo cual, el criterio de seleccionar los documentos que mencionen el vocablo "autos", no parece el más efectivo.

Una mejora posible, es retornar también los documentos que, en lugar de llamar a los automóviles como "autos", los denominan "coches", "carros", etc. Porque el significado de estas palabras es, en esencia, el mismo que "autos": Y esto es lo que realmente importa.

Además, existen temas como "Fórmula 1" o "Medios de transporte" que, si bien no son exactamente lo mismo que el tópico de "autos", tienen relación y al usuario que está buscando sobre "autos" es probable que le interesen: Por tanto, sería de cierta utilidad, presentar también como posibles resultados de interés, enlaces a documentos que mencionan estas palabras.

El W3C, instituciones públicas, instituciones privadas e investigadores en general, han estado trabajando en el desarrollo de herramientas semánticas, algunas para el Web (Web semánticas), que permiten mejorar las búsquedas sobre textos.

Como se puede esperar, todo parte de la catalogación de los textos que se comparten y sobre los que se desean realizar búsquedas.

La tecnología semántica ya tiene cierto recorrido **[14]** y se han propuesto varios estándares y herramientas para su construcción y funcionamiento; pero, ¿son realmente útiles?

El presente trabajo realiza un experimento con algunas tecnologías semánticas para el mejoramiento del sistema de búsquedas y la estandarización de una aplicación de gestión de información: Alejandría.

1.3.- ¿Qué es Alejandría? [5]

Alejandría es una aplicación informática creada hace ya más de 10 años con la idea de gestionar los diferentes tipos de documentos que se manejan en bibliotecas: Monografías, publicaciones seriadas, tesis, etc. La aplicación, logró éxito dentro del mundo bibliotecario y cuenta en la actualidad con una base instalada importante.

Las principales ventajas que tenía Alejandría al momento de su aparición comercial eran varias: Cumplía con los principales estándares bibliotecarios, contaba con interfaz Web (incipiente en aquellos momentos) y su motor de búsquedas.

Posteriormente, el modelo de "tipos de documentos" fue extendido a un modelo de "Tipos de Objetos de Información" que le dio una mayor versatilidad y generalidad al concepto, y permitió que Alejandría se utilizara en la construcción de otros tipos de aplicaciones: Archivos, expedientes de recursos humanos, agendas y actas, gestión de correspondencia, portales Web informativos, sistema de seguimiento de tareas, etc.: Este modelo extendido se llamó "Tecnología de Bases de Información" y tiene aplicación en ámbitos donde se tengan problemas de manejo y gestión de información.

Alejandría es un software para la conceptualización y puesta rápida en funcionamiento de otras aplicaciones que se encarguen de gestionar objetos de información (documentos, en su mayoría). Cuando se realiza una implementación de Alejandría para cierto ámbito, se siguen las fases presentadas en la Figura 1 **[20]**.

Los pasos que se siguen en cada una de las fases del modelo de desarrollo son:

Conceptualización

1. Levantamiento de información
2. Visión
3. Requerimientos
4. Casos de uso
5. Modelo preliminar
6. Estimaciones de tamaño
7. Plan de proyecto
8. Sitio Web del proyecto

Modelado

1. Análisis de procesos de información
2. Modelo de la aplicación en el dominio de la información
3. Revisión y Refinamiento iterativo del modelo
4. Publicación del modelo
5. Validación

Construcción

1. Configuración y personalización usando las herramientas de la plataforma de productos de la Tecnología de Bases de información
2. Revisión del prototipo
3. Revisión del documento automático de descripción del modelo
4. Adecuación
5. Publicación del prototipo
6. Validación

Transición

1. Instalación en el servidor preoperatorio (físico o virtual) de la organización receptora
2. Instalación en el servidor de producción de la organización receptora
3. Soporte técnico de garantía

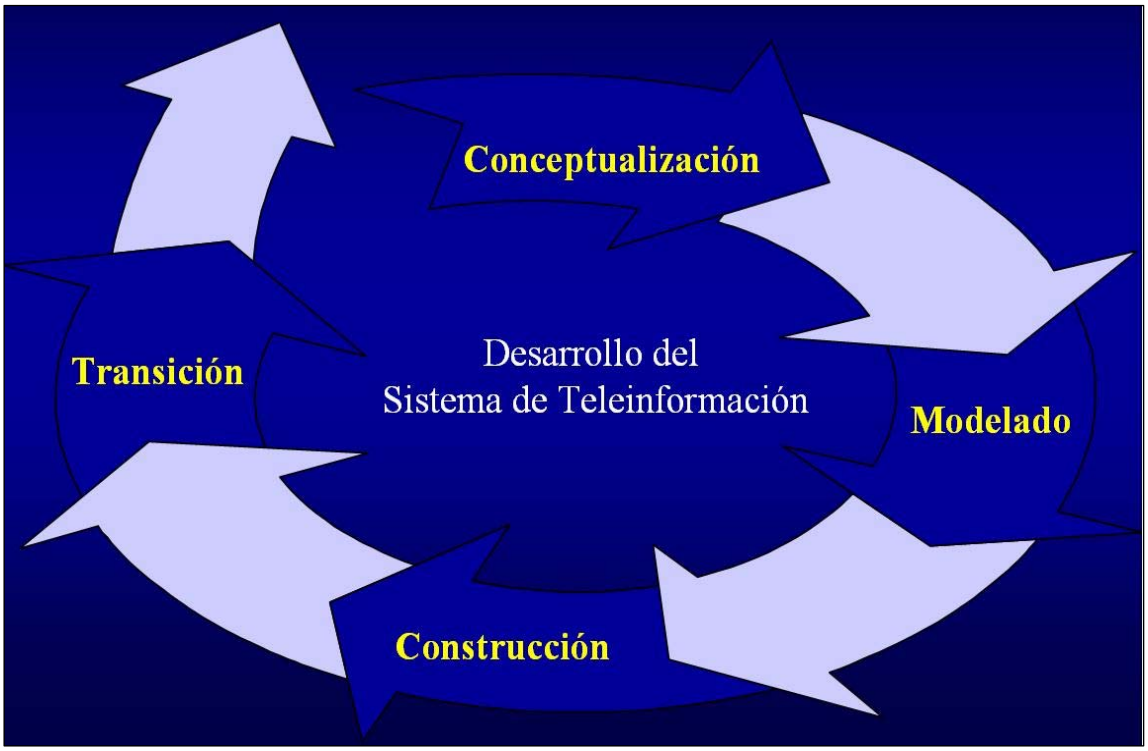


Figura 1: Modelo de desarrollo de aplicaciones utilizado para Alejandría

Al entrar en funcionamiento, el motor de Alejandría permite hacer búsquedas sobre los objetos (documentos) ingresados. En el mejor de los casos, todo este proceso se realiza sin codificar o programar las funcionalidades de gestión de los tipos de objetos de información; ya que estas han sido programadas previamente de forma genérica, y se adaptan a la parametrización y modelado realizado.

La realización de una nueva implementación de Alejandría sigue estos pasos y

también el desarrollo de nuevas características, como el presente trabajo; el cual se enmarca dentro del paso de Adecuación (4) de la etapa de Construcción de la metodología.

1.4.- Ejemplo en Alejandría: ¿Quién ha escrito sobre Manuelita?

Utilizando el portal Web "Luces de Bolívar en la Red" **[8]**, que funciona sobre Alejandría, como ambiente de pruebas, se presenta a continuación el resultado que provee la aplicación cuándo se desea conocer quién ha escrito sobre Manuelita. Posteriormente, en el capítulo 4, se comparará este resultado con el que se obtiene a través de "Alejandría Inteligente"; es decir, a través de la incorporación de las características semánticas a Alejandría.

Alejandría, en la actualidad, no tiene la capacidad de procesar ningún tipo de consulta en lenguaje natural; por tanto, al igual que sucede durante la utilización de un buscador en Internet, el usuario debe pensar cómo convertir la consulta que quiere realizar en descriptores que puedan producir una respuesta lo más acorde posible a lo que él está buscando. En este caso pareciera que simplemente introduciendo la palabra "Manuelita", podemos obtener las publicaciones que tratan sobre Manuelita y de allí extraer los autores que han escrito sobre ella.

En Alejandría, se debe acceder a la pantalla de Búsquedas para la realización de consultas básicas (por descriptores). En la Figura 2 se muestra esta interfaz y se presenta cómo se realiza la consulta sobre Manuelita:

Figura 2: Búsqueda simple de "Manuelita" con Alejandría

A pesar de que existen otros tipos de búsquedas, la búsqueda simple es la que mejor se adapta a lo que se desea, pues las otras búsquedas, permitirían discriminar por tipos de descriptores: Título, autor, etc., que no implican una ventaja cierta (En nuestro ejemplo), pues podría reducirse la cantidad de resultados obtenidos.

Al realizar la consulta, se obtiene el siguiente resultado:

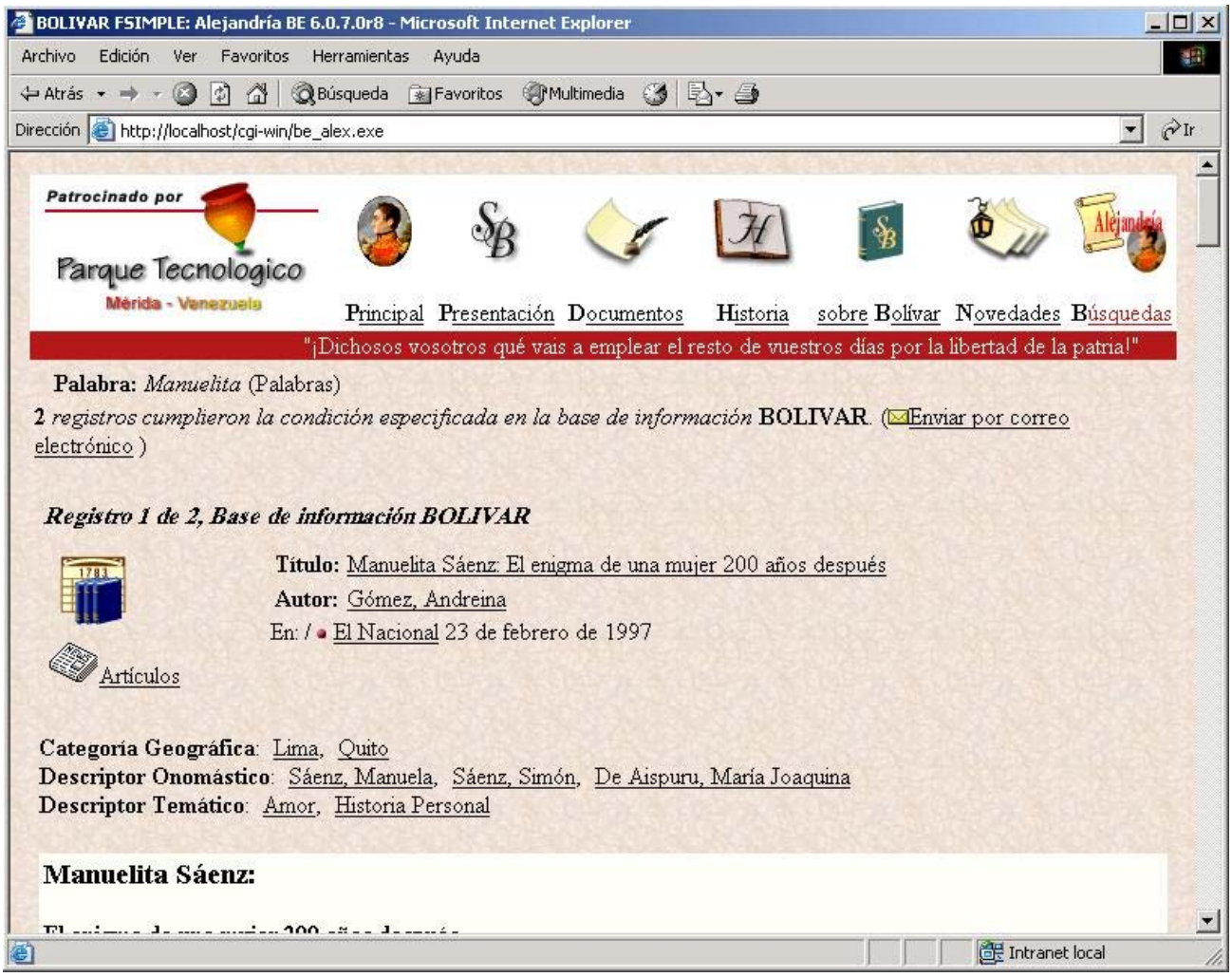


Figura 3: Resultado en Alejandría de la realización de una búsqueda simple con el término "Manuelita"

Como se aprecia en la Figura 3, se obtienen dos registros que tratan sobre Manuelita, el primero lleva por título: "*Manuelita Sáenz: El enigma de una mujer 200 años después*". El otro registro no se puede apreciar en la figura, pero su título es: "*Manuelita Sáenz: una pasión desbocada*".

Este es el tipo de consultas que se suelen realizar en los sistemas gestores de información actuales y es el procedimiento base para los resultados que muestran los buscadores de información de Internet.

1.5.- Un experimento de Web semántica con Alejandría

Las tecnologías Web semánticas se encuentran aún en etapa de consolidación. Prometen grandes beneficios: Como la mejora de las búsquedas generales que se realizan a través de

Internet y la creación de una plataforma para la utilización de piezas de software sofisticadas que lleven a cabo labores de procesamiento de información. Es por ello, que se planteó la necesidad de realizar un trabajo donde estas tecnologías se pusieran a prueba. Se decidió investigar sobre las principales tecnologías Web semánticas que existen, escoger las que se consideraran más útiles y realizar un experimento práctico con estas.

1.5.1.- ¿Por qué usar Alejandría?

Algunas de las razones por las cuales se escogió a Alejandría como gestor de información para la implementación de algunas tecnologías Web semánticas fueron:

a) Es una aplicación de gestión de información donde las búsquedas textuales son intensamente utilizadas.

b) Alejandría es una aplicación estable y en continua mejora (Tiene más de 10 años en el mercado y continuamente se le están agregando mejoras para adaptarla a las necesidades de los usuarios), donde los cambios son consensuados (se discuten en reuniones semanales de desarrolladores) y existe una metodología de desarrollo bien establecida **[20]**.

c) La aplicación cuenta con un gran número de usuarios, con lo cual, el trabajo experimental podría llegar a ser utilizado por un amplio número de personas.

d) Los representantes de la empresa que desarrolla y comercializa Alejandría mostraron disposición a colaborar en un proyecto de estas características, dedicando tiempo y recursos para ello.

2.- El problema general: Análisis automático de significados

Los usuarios de Internet y de aplicaciones de gestión de información utilizan interfaces de búsqueda para poder acceder a lo que quieren leer o a algún texto que los ayude o dé respuesta a sus preguntas. La intención es que estas búsquedas retornen los documentos más relevantes sobre lo que se está buscando. Sin embargo, los resultados de estas búsquedas muchas veces carecen de exactitud, se muestran muchos (demasiados) documentos no bien ordenados por relevancia y la información sólo es procesable por un humano; pues los formatos electrónicos en los que la información es publicada no facilitan a un sistema computacional relacionar la información y poder "conocer" de qué tratan los distintos documentos.

Las búsquedas que se realizan en la actualidad a través de estas interfaces están basadas en la sintaxis de los descriptores que se introducen, donde el verdadero significado (semántica) de lo que

se desea consultar se diluye en la complejidad de nuestro lenguaje y en la inexactitud de los descriptores.

2.1.- De búsquedas basadas en sintaxis a búsquedas basadas en semántica

Las consultas textuales, que realizan las aplicaciones informáticas y los motores de búsquedas en Internet, han ido mejorando con el transcurrir de los años, debido entre otros aspectos, al uso de técnicas estadísticas y a algunas técnicas de indización y búsqueda rápida de patrones sintácticos.

A pesar de estas mejoras, los usuarios se ven en la necesidad de codificar las preguntas, usar operadores lógicos como AND, OR y NOT, dedicar tiempo en el chequeo de textos que, en realidad, no tratan sobre el tema buscado (o que llevan todos a un mismo documento), realizar búsquedas de varias formas (usando posibles sinónimos y agregando cada vez más palabras descriptivas), etc. Todo esto debido a las limitaciones intrínsecas de las búsquedas sintácticas, que se basan simplemente en buscar la aparición de un vocablo y no consideran el significado de lo que se busca (su semántica).

Desde la creación de la Web, Tim Berners-Lee planteó la necesidad de introducir metadatos en los textos publicados electrónicamente para tener mayor capacidad de colaboración entre grupos heterogéneos de distintas partes del mundo (Evidencia de esto son las etiquetas META que existen en las especificaciones del HTML **[1]**). Esta catalogación involucra un trabajo adicional (tiempo y esfuerzo) que lleva a muchos a no realizarlo. Pero, si se desea estar más cerca de la semántica de lo escrito, estos metadatos son indispensables.

Se plantea la necesidad de establecer estándares para definir y compartir la información de los recursos que se desean publicar. Además, la información compartida debe estar en un formato que las computadoras puedan procesar; para que puedan ser empleadas en razonamientos simples, toma de decisiones y recopilación de información, entre otras cosas.

2.2.- Intercambio de información en formato estándar

Desde sus inicios, el HTML (HyperText Markup Language) ha permitido la definición de metadatos **[1]**. Sin embargo, las funcionalidades que ofrece HTML para la especificación de metadatos resultan limitadas y su utilización queda a discreción del desarrollador.

En el mundo de las aplicaciones de escritorio, el problema ha sido mucho más grave, pues muchos programas tienen su propio formato de almacenamiento de la información; y en el

mejor de los casos, si los contenidos son almacenados en texto plano, estos archivos no tienen una estructura estándar; por lo tanto, difícilmente podrán ser procesados y analizados por otra pieza de software, más allá de la sintaxis de las palabras almacenadas.

En 1998, el eXtensible Markup Language (XML) **[19]** se convirtió en una recomendación del W3C para estructurar y compartir datos (describiéndolos). XML es una convención notacional que permite especificar lenguajes como HTML, pero que no está atada a ninguna semántica particular. No es un lenguaje completo, sino un medio para hacer lenguajes, pues los usuarios pueden definir la semántica de las etiquetas. A través de estos años, XML se ha convertido en el estándar de facto para el intercambio de información. Es de hacer notar, que esta estandarización en el formato de intercambio de información es un gran paso en la dirección correcta, pero que no por publicar la información en XML, ésta podrá ser procesada por las computadoras: XML es útil, primordialmente, para estructurar y documentar la propia información.

XML tiene la bondad y el problema de que el implementador o creador es quien escoge el nombre de las etiquetas. Por tanto, si se desea intercambiar información con alguien en XML, se debe llegar a un acuerdo en cuanto a los nombres de las etiquetas que se utilizarán y el significado de cada una. Esta discrecionalidad hace que, por ejemplo, lo que alguien pudiera colocar como <creador>, otra persona pudiera escribirlo como <autor>, dificultando así el intercambio de información y su procesamiento automático.

2.3.- Semántica de las consultas

Si se introduce en Google la consulta:

address hotel hilton paris

Porque se desea conocer dónde está ubicado el Hotel Hilton en la capital francesa, se encontrará que los enlaces de la respuesta hacen referencia, en su mayoría, a un incidente con el cuaderno de direcciones de “Paris Hilton” (hija del dueño de la cadena de hoteles Hilton). El URL de la consulta en Google es:

http://www.google.co.ve/search?hl=es&q=address+hotel+hilton+paris&btnG=B%C3%BAsqueda+en+Google&meta=

Resulta comprensible que se obtengan resultados como este, pues el sistema no advierte la ambigüedad en la consulta: Varias cosas pueden ser representadas a través de las palabras que se introdujeron.

Debido a los sistemas de búsquedas que se han desarrollado y utilizado comúnmente,

los usuarios se han acostumbrado a codificar las preguntas, resumir y omitir palabras para ahorrar tiempo. Esta codificación de la pregunta da muy buenos resultados en el caso de preguntas simples, pero cuando se desean respuestas a algunas preguntas específicas, es cuando comienza a fallar. Por ejemplo, si deseamos dar respuesta a las siguientes preguntas:

- *¿Qué líneas aéreas ofrecen vuelos desde Maracaibo a Caracas?*
- *¿En qué clínicas de Venezuela realizan operaciones de córnea?*

Las codificaciones de estas preguntas, para un buscador de información como Google, podrían ser:

vuelos Maracaibo Caracas

clínica Venezuela operación córnea

Los URL, para Google, de estas consultas son respectivamente:

http://www.google.co.ve/search?hl=es&q=vuelos+Maracaibo+Caracas&btnG=B%C3%BAsqueda&meta=

http://www.google.co.ve/search?hl=es&q=Cl%C3%ADnica+Venezuela+operaci%C3%B3n+c%C3%B3rnea&btnG=B%C3%BAsqueda&meta=

En los resultados que se obtienen se pueden apreciar varios problemas graves:

a) La codificación de la pregunta lleva a una ambigüedad indescifrable para el software tradicional. El computador no puede captar qué se desea exactamente.
b) En la Internet existe actualmente información suficiente como para dar una respuesta aproximada a estas preguntas. Pero para el computador, resulta prácticamente imposible porque la información no se encuentra identificada y catalogada debidamente para que un programa la pueda procesar y pueda identificar relaciones y realizar inferencias. La información en la Web está almacenada en infinidad de formatos y primordialmente en las estructuras de los lenguajes naturales.

En respuesta al problema a) caben las preguntas: ¿Existe una forma más precisa que plantear la pregunta en lenguaje natural? ¿Qué recursos se pueden usar para ayudar al usuario a plantear una consulta no ambigua o lo menos ambigua posible para el computador?

Del problema b) se puede ver claramente que la información compartida debe ser preprocesada y estructurada para que un computador pueda dar mejores respuestas.

3.- Solución al problema general de análisis automático de significados

Para que un sistema computacional pueda tomar en cuenta la semántica de la información publicada, quien hace pública esta información debe catalogarla; es decir, especificar de qué trata, quién fue su creador, fecha de publicación, con qué otros temas se relaciona, etc.

Además, esta catalogación no puede ser en un formato personalizado, sino que debe ser realizada siguiendo algún estándar de catalogación que haga universal el trabajo de catalogación realizado.

Una vez catalogados los documentos publicados en un formato estándar, los motores de búsqueda deben hacer uso de tecnologías que permitan el aprovechamiento de esta catalogación; algunas de estas tecnologías y estándares se plantean a continuación como una posible arquitectura para el análisis automático de significados.

3.1.- Metadatos y Dublin Core

Todo documento compartido en Internet o a través de una aplicación, debería contar con metadatos que faciliten su clasificación y búsqueda. Además, los metadatos utilizados deberían ser estándares, para que otras aplicaciones o usuarios puedan entender exactamente a qué se refiere cada uno de ellos, se pueda realizar el intercambio de información de una forma estándar y se establezca un piso para las búsquedas basadas en semántica.

El Dublin Core es un conjunto de 15 descriptores o metadatos, definidos en 1995 por la Online Computer Library Center **[10]** (Una agrupación de bibliotecas) y otras instituciones, en la ciudad de Dublin (Ohio, USA); con la idea de ser la información mínima que todo documento compartido debería tener. Los 15 elementos básicos que conforman el Dublin Core y su semántica son:

Title: El nombre dado al recurso, usualmente por el creador o publicador. Puede ser el mismo título del recurso o algo más descriptivo.

Author o Creator: Persona u organismo principal responsable por la creación del contenido intelectual del recurso. Es decir, autores en el caso de documentos escritos, artistas, fotógrafos, etc. En el caso de recursos visuales.

Subject: El tema del recurso. Comúnmente, será expresado como palabras claves o frases que describen el tema o contenido del recurso. Se utilizan vocabularios controlados y esquemas de clasificación formal.

Description: Una descripción formal del contenido del recurso, incluyendo abstracts en el caso de

objetos tipo documento o descripciones del contenido en el caso de recurso visuales.

Publisher: La entidad (Es decir agencia, incluyendo unidad, sucursal, sección) responsable de hacer al recurso disponible en su forma actual, tal como una casa de publicación, el departamento de una universidad o una entidad corporativa.

Contributor: Persona u organización no especificada en el elemento Creator que ha realizado una contribución intelectual significativa al recurso, pero cuya contribución es secundaria a cualquier persona u organización especificada en el elemento Creator. Por ejemplo: Editor, transcriptor, ilustrador.

Rights: Una sentencia sobre los derechos de autor o un identificador que enlaza o direcciona a una sentencia que describe los derechos de autor.

Date: Una fecha asociada con la creación o disponibilidad del recurso.

Type: La categoría del recurso; como novela, poema, página Web, reporte técnico, ensayo, diccionario.

Format: El formato de la data del recurso, usado para identificar el software y posiblemente el hardware que pudiera ser necesitado para desplegar u operar el recurso; Por ejemplo: Postscript, HTML, texto, JPEG, XML.

Identifier: Una cadena de caracteres o número usado para identificar de forma única al recurso. Ejemplos para recursos de la red incluyen URLs, Purls y URNs. ISBNs u otros nombres formales también pueden ser utilizados.

Source: El trabajo, ya sea impreso o en electrónico, del cual se deriva el recurso. Este elemento no se aplica si el recurso se encuentra en su forma original.

Language: El lenguaje del contenido intelectual del recurso.

Relation: Relación con otros recursos. Por ejemplo: Imágenes en el documento, capítulos en un libro, ítems de una colección.

Coverage: Localización espacial y duración temporal característica del recurso.

La semántica de cada uno de estos descriptores fue definida y es mantenida por la organización internacional en que se ha convertido el Dublin Core **[6]**. Los metadatos Dublin Core pueden ser utilizados varias veces dentro de un mismo texto, en cualquier orden y no es necesario utilizarlos todos.

Los metadatos Dublin Core no fueron creados con la idea de que sean los únicos que se utilizan en un documento electrónico compartido. Se idearon con la intención de que sean

utilizados por todos los documentos compartidos en Internet, sirviendo de conjunto común entre documentos pertenecientes incluso a distintos dominios. Además, se promueve la definición de más metadatos para dominios específicos, que ayuden a estructurar y documentar la información, siempre y cuando se parta, como conjunto base, de los metadatos definidos por Dublin Core.

El Dublin Core se ha convertido en un estándar de facto; por lo que toda aplicación o sitio web que comparta documentos debería incluir estos metadatos.

3.2.- Relaciones entre piezas de información

El W3C basa gran parte del esfuerzo de desarrollo del concepto de Web Semántica en el Resource Description Framework **[4]** y el Resource Description Framework Schema **[24]**. Estas tecnologías, basadas en XML, permiten definir y catalogar recursos a través de su relación con otros, siempre que sean identificables a través de URIs[1] (Uniform Resource Identifiers) en Internet.

3.2.1.- El Resource Description Framework (RDF)

El RDF es un lenguaje, basado en XML, desarrollado por el W3C para representar información sobre recursos en Internet. El RDF permite establecer relaciones funcionales entre recursos Web y describir estos recursos.

El RDF se basa en tripletas: Cuando alguien construye una sentencia, generalmente la puede articular en tres partes: sujeto, predicado y objeto (un grupo de estos elementos es conocido como tripleta). Un ejemplo de tripleta es: El carro (Sujeto) es de color (Predicado) azul (Objeto).

El código fuente de un archivo RDF de ejemplo es:

```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">
      <rdf:Description rdf:about="http://www.solsinca.com">
            <dc:title>Soluciones en Sistemas Informáticos C.A.</dc:title>
            <dc:author>Icaro Alzuru C.</dc:author>
      </rdf:Description>
</rdf:RDF>
```

Este ejemplo muestra que el recurso http://www.solsinca.com tiene como título Soluciones en Sistemas Informáticos C.A. y su autor es Ícaro Alzuru C. También se observa en la línea 3, que se están utilizando los metadatos Dublin Core: Referenciados a través del prefijo “dc”. Por

[1] URI (Uniform Resource Identifier, identificador uniforme de recursos): Texto corto que identifica unívocamente cualquier recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia ...) accesible en una red **[15]**

ello, en las líneas 5 y 6 se encuentran los metadatos title y publisher de Dublin Core. Debido a que se conoce este conjunto de metadatos es que se puede estar seguro que dc:title se refiere al título o nombre principal y que dc:author es el autor o creador del recurso.

Los archivos RDF hacen referencia a tipos de recursos definidos en espacios de nombres. El **espacio de nombres [2]** es un documento XML qué define los tipos de elementos y nombres de atributos que pueden contener otros documentos XML, identificándolos a través de un URI. En el ejemplo anterior, el archivo está utilizando dos espacios de nombres: http://www.w3.org/1999/02/22-rdf-syntax-ns# y http://purl.org/dc/elements/1.1/. El espacio de nombres permite al lector conocer qué significa el metadato al cual se hace referencia en un archivo. Así, en la dirección http://purl.org/dc/elements/1.1/ se encuentra la especificación de los metadatos Dublin Core.

3.2.2.- El Resource Description Framework Schema (RDFS)

RDFS es un lenguaje basado en XML para la especificación de vocabularios de RDF; es una extensión semántica de RDF, que provee mecanismos para describir grupos de recursos relacionados y las relaciones entre estos recursos. Los recursos en RDFS son divididos en grupos llamados clases. Los miembros de una clase son conocidos como *instancias* de la clase, la clases mismas son recursos, normalmente identificadas por un URI y descritas usando propiedades RDF.

El RDFS sirve para definir clases y subclases de metadatos; generando un diagrama de clases como en la programación orientada a objetos. En el RDFS existen elementos como la herencia de propiedades de una clase a sus subclases. El RDFS es quien contiene las definiciones que posteriormente se usan en archivos RDF: Los archivos RDF son la información con sus metadatos, mientras que el archivo RDFS contiene definiciones de metadatos. El archivo RDF hace entonces referencia al archivo o archivos RDFS donde están los metadatos que utiliza.

Los archivos RDFS son especies de recopilaciones (interrelacionadas) de metadatos de metadatos. Un pequeño ejemplo de un archivo RDFS es el siguiente:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs =
"http://www.w3.org/2000/01/rdf-schema#" xml:base =  "http://www.animales.com/animales">
        <rdfs:Class rdf:ID="mamiferos" />
        <rdfs:Class rdf:ID="caballo">
            <rdfs:subClassOf rdf:resource="#mamiferos"/>
        </rdfs:Class>
        <rdfs:Class rdf:ID="perro">
            <rdfs:subClassOf rdf:resource="#mamiferos"/>
        </rdfs:Class>
</rdf:RDF>
```

El ejemplo de RDFS explica que las clases caballo y perro son subclases de la clase mamíferos. Esta clase de relaciones taxonómicas, es tan frecuente en la computación, que se ha comenzado a hablar de un término prestado de la Filosofía para referirse a ellas: **Ontologías**.

3.3.- Las ontologías y el Web Ontology Language (OWL)

En computación, una **ontología** es una conceptualización de un dominio; es decir, la definición detallada de los objetos que existen en cierto ámbito y de las relaciones que hay entre estos.

A efectos de poder manipular información, si se cuenta con un conjunto de documentos (o recursos en general) que tratan sobre un determinado tema, resultaría útil contar con la ontología del dominio en estudio; sobre todo si la ontología se define en un formato que pueda ser procesado por el computador.

Precisamente, el Web Ontology Language (OWL) es un lenguaje, creado por el W3C, para la definición de ontologías; utilizando un formato que el computador puede procesar: RDF. El OWL establece un vocabulario, en RDF, para la definición de ontologías. Parte del vocabulario agregado por OWL para la descripción de propiedades y clases es: Tipos de relaciones entre clases (Por ejemplo: Disyunciones), cardinalidad (Por ejemplo: “exactamente uno”), igualdad, un conjunto de tipos de propiedades ampliado, características de propiedades (Por ejemplo: Simetría), clases enumeradas, etc. **[25]**.

El OWL resulta importante porque estandariza la forma en que se definen objetos y relaciones para un dominio particular. Además, en un formato que permite un procesamiento automático y que por tanto, da cabida a que se puedan programar piezas de software para realizar razonamiento sobre los objetos que tratan un conjunto de recursos.

El Dublin Core, mencionado anteriormente, establece un conjunto mínimo de metadatos para los recursos compartidos. El RDF especifica una forma de describir y relacionar los recursos en Internet. Dentro de RDF se pueden utilizar metadatos Dublin Core para dar información sobre un recurso.

El OWL va un poco más allá, y define un vocabulario, sobre RDF, para la especificación de los objetos y sus relaciones en un dominio en particular (ontologías). El OWL también permite indicar que cierta propiedad es equivalente a un metadato Dublin Core específico; por lo que se pueden usar en conjunto. Esto se puede apreciar en el siguiente extracto de un archivo OWL:

```
<rdf:RDF
   xmlns:dcterms="http://purl.org/dc/terms/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://purl.org/dc/elements/1.1/"/>
  </owl:Ontology>
. . .
  <owl:DatatypeProperty rdf:ID="Nombres">
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
   <dc:creator rdf:datatype="http://www.w3.org/2001/XMLSchema#string"></dc:creator>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
. . .
```

Como se puede apreciar en las líneas resaltadas en negrita del cuadro anterior; para especificar que una Propiedad es equivalente a un metadato del Dublin Core, se importa el Dublin Core, se establece un espacio de nombres para sus términos y se indica, en la propiedad, a qué metadato del Dublin Core equivale.

El OWL fue diseñado para codificar aplicaciones ontológicas en diversos dominios de conocimiento. Hay razones para creer que un lenguaje especializado en un dominio, dará mejores resultados. Por esa razón, se incorporó un mecanismo ontológico estrechamente ligado al procesamiento de los lenguajes naturales: El sistema WordNet **[18]**.

3.4.- La lingüística computacional y WordNet

Una ontología, a través de las relaciones, establece un grafo entre los tipos de objetos que existen en un dominio dado. Al definir una ontología, por ejemplo con OWL, se cae en la trampa de bajar del terreno de la semántica al terreno de la sintaxis (Esto es inevitable, si se quiere seguir usando el lenguaje): Es decir, se debe representar un significado a través de una palabra. Esto tiene algunas consecuencias; por ejemplo, si se establecen relaciones del tipo:

Los autos tienen ruedas

Los autos tienen frenos

Los autos tienen volante, etc.

Se pudiera construir un programa que retorne las partes que conforman a los autos y que enlace con portales Web de empresas que comercializan estas partes.

Pero, por ejemplo, ¿qué ocurre con una empresa que no menciona que comercializa

ruedas; sino las denomina “cauchos”, “neumáticos” o “llantas”?: Un sistema de búsquedas sintáctico no retornará el enlace a la página de esta empresa como opción. Así como tampoco mostrará los enlaces de sitios que traten sobre rines, tren delantero, suspensión, etc.

¿Cómo conseguir, al momento de realizar una búsqueda, estas palabras relacionadas al concepto o tema que introdujo el usuario?: Una solución es la tecnología **WordNet**.

WordNet es un sistema electrónico de referencia léxica, desarrollado por el Cognitive Science Laboratory de la Universidad de Princeton **[18]**. Se pudiera decir que WordNet es un diccionario semántico, pues se define un significado y se establecen las palabras que lo representan, este conjunto de palabras se denomina *synsets*. Para un significado o synset dado, se establecen relaciones con otros significados (synsets). Los tipos de relaciones, más importantes, que existen entre synsets son (entre otras) **[17]**:

- **Hiperónimos**: Y es un hiperónimo de X si todo X es un (tipo de) Y
- **Hipónimos**: Y es un hipónimo de X si todo Y es un (tipo de) X
- **Holónimos**: Y es un holónimo de X si X es una parte de Y
- **Merónimos**: Y es un merónimo de X si Y es una parte de X

A través de estas relaciones, se esconden gran cantidad de aplicaciones que se le pueden dar a la WordNet. WordNet se puede considerar una ontología general.

La implementación de la WordNet en inglés, creada por la Universidad de Princeton **[18]**, es Open Source: Se puede modificar, compilar y está disponible con un API en C para poder ser accedida programáticamente. Para el lenguaje español, se creó una WordNet, a través del proyecto EuroWordNet **[3]**, pero hay que pagar para obtenerla, así como por sus programas de creación y consulta.

3.5.- Representando la semántica de las consultas usando Prolog

Prolog es un lenguaje de programación lógica que ha probado su utilidad para la representación de conocimiento, especialmente conocimiento vertido en forma declarativa o en una combinación de declaraciones (afirmaciones y negaciones) y código imperativo. Es además muy apropiado para hacer prototipos rápidamente **[12]**. Una de las primeras aplicaciones de Prolog, fue la representación de gramáticas de lenguajes naturales **[30]**, esfuerzos que culminaron con un macro-lenguaje para representar reglas de producción (reglas gramaticales); conocido como DCG (Definite Clause Grammars) que está incluido en el Standard Prolog.

Como se deduce del punto 2.3; la utilización de consultas en lenguaje natural, permite

obtener mayor información sobre lo que el usuario desea, y por tanto, podría dar una respuesta más exacta, sobre todo cuando no son consultas generales.

A través de la utilización de Prolog, se pueden reconocer tipos de preguntas. Este reconocimiento de patrones posibilita dar una respuesta más acertada a la pregunta. Se deben establecer cuáles son los tipos de preguntas más importantes que se desean reconocer y programar en Prolog su identificación.

Es necesario acotar, que el uso de consultas en lenguaje natural se plantea como una alternativa a las consultas a través de los términos que se utilizan comúnmente. Para consultas genéricas sobre un tema, la forma tradicional de consulta parece ser suficiente. La utilidad de la consulta en lenguaje natural dependerá de lo específico de la pregunta.

4.- La solución en un experimento particular: Extensiones al sistema de consulta documental Alejandría

En Hacer Sistemas, compañía desarrolladora de Alejandría, se tiene un proceso bien definido de desarrollo y construcción de nuevas características para Alejandría. El presente proyecto siguió, durante un año esta metodología para la consecución del trabajo final. Este proceso incluyó reuniones semanales con el grupo de desarrolladores de Alejandría para validar lo que se estaba haciendo. El trabajo realizado se enmarca en la etapa de Adecuación de la fase de Construcción de la Metodología Alejandría para el Desarrollo de Sistemas de Teleinformación **[20]**. Los pasos de esta metodología, que fueron seguidos durante el desarrollo de esta nueva característica (Funcionalidad) para Alejandría, fueron:

Conceptualización
- Registro de solicitud
- Análisis y aceptación
- Planificación
- Acuerdo

Modelado
- Diseño
- Acuerdo

Construcción
- Codificación
- Pruebas unitarias
- Validación interna e Integración
- Pruebas de integración
- Validación

Transición

- Generación de instalador
- Documentación interna (Ayuda en línea)
- Documentación externa
- Certificación

Las tecnologías aquí propuestas como solución particular para Alejandría (OWL, RDF, XML, WordNet, Prolog), pueden ser empleadas por cualquier sistema de gestión de información. La utilización de estas tecnologías en Alejandría respondió sobre todo, al deseo de realizar un trabajo práctico y funcional, que permitiera mostrar y probar posibles respuestas a las interrogantes planteadas en este estudio.

4.1.- El sistema de consultas de Alejandría

Cuando se está realizando una conceptualización de una aplicación en Alejandría, se definen los tipos de objetos de información y se establecen sus campos o características. En esencia, estos campos pueden ser de tres tipos: Predefinidos, variables adicionales y descriptores. Los campos que entran dentro de los grupos predefinidos y descriptores, permiten que se puedan realizar búsquedas sintácticas a través de ellos. Además, el definir un campo como descriptor, implica que se puede realizar una catalogación de los objetos de información a través de los valores de este campo.

Internamente, Alejandría divide el contenido de los campos predefinidos o descriptores en palabras y registra la ocurrencia de estas palabras. Esto acelera el procesamiento de las búsquedas que realizan los usuarios y permite describir algunas características de los documentos compartidos.

La interfaz de Alejandría para la realización de consultas permite realizar búsquedas simples, sobre los campos principales del conjunto de objetos de información, búsquedas avanzadas combinando metadatos y operadores lógicos, y exploraciones sobre los contenidos de tablas de metadatos. La interfaz Web de Alejandría para consultas puede apreciarse en la Figura 4:

Figura 4: Forma o interfaz de búsquedas generales en Alejandría

Como se puede apreciar en la Figura 4, Alejandría permite combinar, en un momento dado, hasta 3 criterios de búsqueda, donde cada uno de ellos permite consultar por los distintos campos predefinidos y descriptores. La consulta ingresada puede ser realizada como palabra, comienzo de frase o frase completa. Se observa además, que se pueden combinar criterios de búsqueda a través de los operadores lógicos “y”, “o” e “y no”.

Los descriptores en Alejandría son, en esencia, metadatos (Metadato: Dato estructurado que describe las características de un recurso de información); pues describen al objeto y permiten catalogarlo y hacer búsquedas a través de este campo.

Este ambiente de consultas y el hecho de que se pueden establecer descriptores para la catalogación de los objetos de información, hacen que Alejandría tenga más características para las búsquedas de información que muchos otros programas que gestionan información.

A pesar de estas características avanzadas que presenta Alejandría, el carácter sintáctico de sus búsquedas, deja espacio para mejorar significativamente su proceso de búsqueda mediante la introducción de conceptos y herramientas de la Web Semántica.

4.2.- Diseño Web semántico para Alejandría

Existen muchas iniciativas, estándares y herramientas con intención semántica, que se han desarrollado en los últimos años: XML, RDF, RDFS, OWL, SPARQL, DAMN+OIL, Dublin Core, WordNet, EuroWordNet, etc.

Como se ha venido explicando en el presente documento, se consideró que las tecnologías que pueden ayudar más a la mejora del sistema de búsquedas en una aplicación de gestión de información como Alejandría, son: OWL (XML y RDF), Dublin Core, WordNet y una interfaz que permita introducir las consultas en lenguaje natural (usando Prolog), para poder realizar una mejor aproximación a la semántica de la pregunta.

La arquitectura del sistema propuesto se puede apreciar en la Figura 5. Se muestra rellenada o de color la parte del sistema que ya existía en Alejandría, con fondo blanco, la parte desarrollada. El funcionamiento del sistema es el siguiente:

1. El usuario invoca o accede a la interfaz de consultas semánticas (Forma de consultas en lenguaje natural)
2. El usuario introduce una consulta en lenguaje natural (español) y presiona el botón de buscar Alejandría pasa la consulta al Interpretador de lenguaje natural y accede al archivo consulta.pl (con los patrones de preguntas). Si no puede identificar la consulta, se genera un mensaje de error. Si la consulta es identificada, Alejandría conocerá si se está buscando un campo específico: Autor, tema, etc. y el nombre o tópico buscado.
3. Alejandría accederá a la WordNet temática para buscar los sinónimos e hiperónimos del nombre o tópico consultado. Se generan las listas de sinónimos e hiperónimos.
4. Alejandría realiza 3 búsquedas, consulta con: El término (simple o compuesto) que introdujo el usuario, los sinónimos y los hiperónimos.
5. Los resultados para la consultas son mostrados en la interfaz de respuesta para el usuario. Indicando si el resultado es en base al término original, sus sinónimos o hiperónimos.

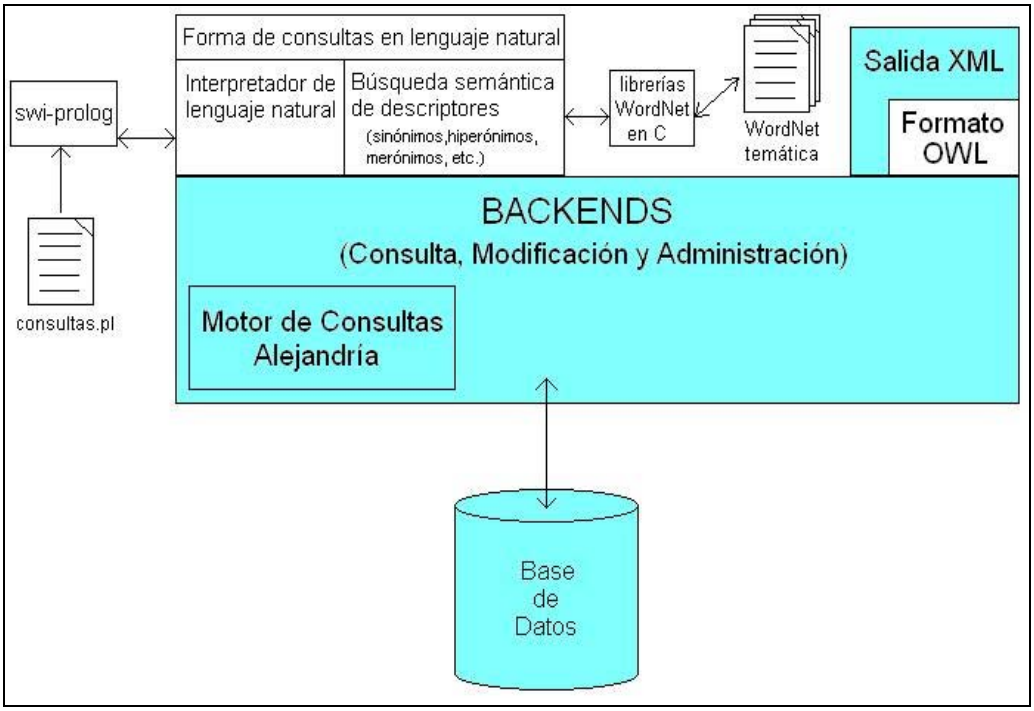


Figura 5: Arquitectura de "Alejandría Inteligente" (Sistema propuesto)

El siguiente diagrama muestra las actividades que se realizan durante la búsqueda semántica:

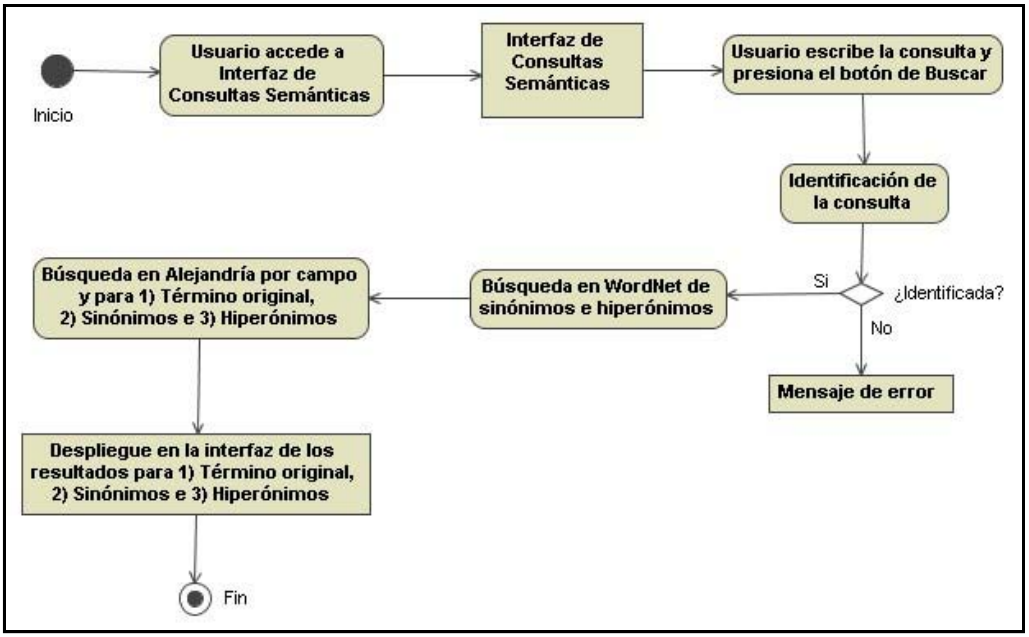


Figura 6: Diagrama de actividades en UML que muestra el flujo de actividades de un usuario Para la realización de una consulta con la interfaz re Consultas Semánticas

4.3.- Ontología Alejandría (Acotación a Monografías)

Por cuestiones de tiempo, para el presente proyecto, se decidió acotar el trabajo al tipo de documento Monografía, que es uno de los tipos de objetos que maneja Alejandría. Si bien existen otros tipos de documentos predefinidos en la aplicación, las monografías son muy utilizadas y cubren gran parte de la funcionalidad de la aplicación. Según el Manual de Alejandría y el Manual de Procedimiento No. 1 del Sistema de Información Bibliográfica: Uso de hojas de trabajo (HDBC y HAC) y tarjeta de registro bibliográfico (TRB), de la Comisión Económica para América Latina (CEPAL), una monografía puede definirse de la siguiente forma:

"*... se considera monografía toda publicación que, reproducida por medios mecánicos, constituya una unidad en sí misma, tenga editorial responsable de su publicación, tenga tapas (aunque no necesariamente duras) y tenga una portada con los datos esenciales para su identificación (autor, título, editorial, lugar y fecha de publicación). Una monografía puede estar constituida por uno o más volúmenes. Cada edición separada de una serie monográfica es también considerada monografía. Se exceptúan de esa categoría los documentos dactilografiados, considerados documentos no convencionales, y las tesis*".

Alejandría contaba con un generador de salidas XML; es decir, un usuario podía ver la información de una monografía consultada en formato XML. Por ejemplo:

```
<?xml version="1.0" encoding="windows-1252" ?>
- <ConsultaAlejandria>
  <BaseInformacion>bolivar</BaseInformacion>
  <BackEnd>/cgi-win/be_alex.exe</BackEnd>
  <VersionBackEnd>Back-end Alejandría BE 5.5.1.9r</VersionBackEnd>
  <Consulta>Acceso=T011900001009/0&nombrebd=bolivar&tsalida=x</Consulta>
  <TotalRegistrosConsulta>1</TotalRegistrosConsulta>
  <NroRegistrosPagina>10</NroRegistrosPagina>
- <Registro>
  <Acceso>T011900001009/0</Acceso>
  <NroOrdinalRegistro>1</NroOrdinalRegistro>
  <TipoDocumento>M</TipoDocumento>
  <TipoReferenciaAnalitica>-</TipoReferenciaAnalitica>
  <Nombre>-</Nombre>
  <Cota>Bol015</Cota>
- <Autor>
  <Nombre>Polanco Alcántara, Tomás</Nombre>
  <Email /><Url />
  <CodigoTipo>P</CodigoTipo>
  <Tipo>Principal</Tipo>
  </Autor>
  <Titulo>Simón Bolívar</Titulo>
  <SubTitulo>Ensayo de una interpretación Biográfica a través de sus documentos</SubTitulo>
  <Fecha>1997</Fecha>
  </Registro>
  </ConsultaAlejandria>
```

Si bien esto añadía cierta estandarización en la salida de Alejandría, los nombres de las etiquetas XML no obedecían ningún criterio escrito que el usuario pudiera consultar. Es decir, no correspondían a ningún estándar predefinido para metadatos. Por ello, se decidió que la salida XML utilizara los metadatos establecidos por Dublin Core. Esto, sin embargo, no es suficiente, pues el Dublin Core no especifica el concepto Monografía. Para que otros usuarios y otro software puedan "entender" los metadatos de la salida XML para una Monografía, esta salida debe regirse por una ontología que esté disponible vía Web a los usuarios de Alejandría y que pueda ofrecerse como una extensión del Dublin Core para otros sistemas que almacenen monografías. Durante el proceso de investigación, se buscó alguna ontología ya definida para el campo de las monografías, pero no se consiguió nada parecido a la implementación de Alejandría del tipo de objeto Monografía. Por tanto, se decidió crear una ontología propia para el dominio de las monografías utilizando OWL.

Se prefirió utilizar OWL en lugar de RDFS debido a que OWL fue concebido como una mejora o extensión de las capacidades de RDF y RDFS; al añadir, entre otras cosas, relaciones entre clases (incluyendo disyunciones), cardinalidad, igualdad, un manejo de propiedades de tipos extendido, características para las propiedades (como simetría), clases enumeradas, etc. El W3C recomienda que si se van a realizar tareas de razonamiento sobre la información compartida por los documentos, el lenguaje debe ir más allá de la semántica básica del RDFS: OWL **[25]**.

Para la creación de la ontología con OWL, se utilizó el programa Protègè en su versión 3.0 **[13]**: El cual es un editor de ontologías creado en la Universidad de Stanford (USA). Las clases o tipos de objetos incluidos para la descripción de una monografía se pueden apreciar en la Figura 7. La ontología creada en OWL usando Protègè, se puede consultar en http://www.alejandria.biz/ontologia/alejandria.owl. No se añadió como anexo al presente trabajo debido a su extensión.

Una vez definida y creada la ontología, se codificó en Alejandría lo necesario para que tuviera un nuevo subtipo de salida que se rigiera por la ontología creada. Un ejemplo de un registro monográfico que sigue la definición ontológica es:

```
<?xml version="1.0" encoding="windows-1252" ?>
- <rdf:RDF xmlns="http://www.alejandria.biz/ontologia/monografias.owl#"
        xmlns:alejandria="http:// www.alejandria.biz /ontologia/alejandria.owl"
xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xml:base="http://www.alejandria.biz/ontologia/monografias.owl">
```

```
- <owl:Ontology rdf:about="">
        <owl:imports rdf:resource="http://www.alejandria.biz/ontologia/monografias.owl" />
  </owl:Ontology>
- <alejandria:Monografia rdf:ID="T011900001009/0">
- <alejandria:Autores>
- <alejandria:Autor rdf:ID="AutoPolanco Alcántara, Tomás">
        <alejandria:Nombre rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Simón
Bolívar</alejandria:Nombre>
        <alejandria:Email rdf:datatype="http://www.w3.org/2001/XMLSchema#string" />
        <alejandria:Url rdf:datatype="http://www.w3.org/2001/XMLSchema#string" />
        <alejandria:NacimientoMuerte rdf:datatype="http://www.w3.org/2001/XMLSchema#date" />
        <alejandria:CodigoTipo
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">P</alejandria:CodigoTipo>
        <alejandria:TipoAutor rdf:datatype="http://www.w3.org/2001/XMLSchema#string" />
  </alejandria:Autor>
  </alejandria:Autores>
  <alejandria:Cota rdf:datatype="http://www.w3.org/2001/XMLSchema#string">980-07-2259-
9</alejandria:Cota>
- <alejandria:Titulos>
      - <alejandria:Titulo rdf:ID="">
            <alejandria:TipoTitulo
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Bol015</alejandria:TipoTitulo>
            <alejandria:Nombre
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">33</alejandria:Nombre>
        </alejandria:Titulo>
        <alejandria:SubTitulo>Ensayo de una interpretación Biográfica a través de sus documentos
        </alejandria:SubTitulo>
  </alejandria:Titulos>
  <alejandria:Colacion rdf:datatype="http://www.w3.org/2001/XMLSchema#string">M
  </alejandria:Colacion>
  </alejandria:Monografia>
</rdf:RDF>
```

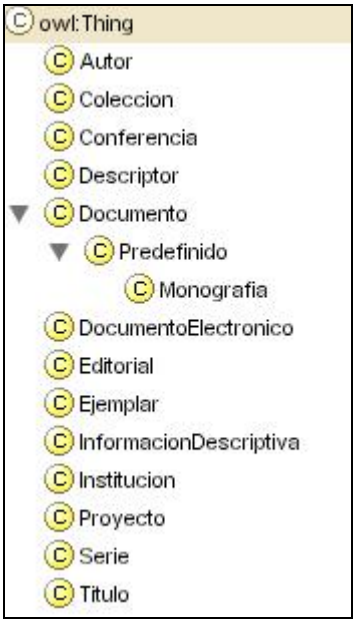


Figura 7: Clases descritas en OWL utilizando Protégé

En la ontología desarrollada, se agregaron comentarios (especie de alias) necesarios

para reconocer los metadatos Dublin Core que tenían equivalente en el ámbito monográfico. Esto con la intención de estandarizar más la salida OWL y permitir que cualquier usuario o software pueda entender al menos, si no trabaja con OWL, los metadatos Dublin Core.

Para que una persona o software que lee un registro que sigue la ontología, pueda entender a qué se refiere cada uno de los metadatos definidos, la ontología está disponible públicamente en la dirección www.alejandria.biz/ontologia/alejandria.owl

Con esto, se logró que Alejandría tenga una salida Web en formato XML, y que esta salida siga una ontología que los usuarios pueden consultar y saber exactamente a qué se refiere cada uno de los metadatos del documento, mejorando la estandarización de la salida.

4.4.- Captura de la semántica de la pregunta con Prolog

El procesamiento automático del lenguaje natural es un gran desafío. No existe ningún sistema completamente automático que pueda identificar cualquier oración o expresión idiomática en un lenguaje natural, como haría cualquier hablante nativo de ese lenguaje. Para abordar esa complejidad, se optó por identificar 6 tipos de preguntas modelo comunes, asociadas con las consultas a un sistema como Alejandría. Con respecto a esos 6 tipos de preguntas, se definió una gramática computacional que puede identificar variantes que, una persona interesada en hacer consultas de este tipo, pudiera producir. Todo esto como parte de un esfuerzo por "controlar" la complejidad del lenguaje natural, circunscribiéndolo a cierto tipo de interacciones.

La "Forma de consultas en lenguaje natural" mencionada en la Figura 5, involucró la creación de una Forma (Interfaz gráfica en HTML) a través de la cual el usuario puede realizar consultas en lenguaje natural, esta interfaz se aprecia en la Figura 8. Para la identificación de estas consultas se utiliza el programa Prolog (swi-prolog), implementando la gramática para consultas que se programó. Las consultas modelo programadas para que el sistema pueda identificar son:

1. ¿Quién ha escrito sobre Esto?

2. ¿Dónde consigo información sobre Esto?

3. ¿Cómo hago Esto?

4. ¿Qué es Esto?

5. ¿A quién consulto sobre Esto?

6. ¿Esto es parte de Qué?

Las palabras en mayúsculas dentro de la pregunta (no al principio) son variables o "huecos" que se completan en cada caso con los términos específicos que introduzca el usuario. Por

ejemplo, "Esto", en las primeras 5 preguntas modelos, representa el tema sobre el que el usuario desea averiguar información. El programa Prolog utilizado para la identificación de las consultas se puede apreciar en el ANEXO A.

Con esto, se logró que algunas preguntas puedan ser introducidas en lenguaje natural, que el sistema las identifique y dé una respuesta más exacta.

Si se desea que Alejandría reconozca otros tipos de consultas; se debe programar el reconocimiento de estas en Prolog y agregarlo en el archivo consulta.pl. Además, se debe modificar el módulo de búsqueda semántica de Alejandría para que, una vez sea reconocido este nuevo tipo de consulta, se realice la consulta Alejandría más apropiada para dar respuesta a esta.

4.5.- Mejora de los resultados de búsquedas con WordNet

Para cada dominio donde se implemente Alejandría, se debe crear una WordNet que establezca las definiciones y relaciones de los objetos del dominio.

Una de las aplicaciones Web más famosa desarrollada con Alejandría es "Luces de Bolívar en la Red" **[8]**, sitio Web dedicado a la vida y obra de Simón Bolívar, El Libertador. Para realizar una demostración de la funcionalidad y ver la factibilidad del sistema propuesto, se creo una WordNet sobre Bolívar, teniendo en cuenta la información disponible en el sitio Web mencionado.

La utilidad de la creación de esta WordNet, es que sirve como diccionario de conceptos relacionados para los términos de búsqueda del usuario. Es decir, la WordNet sobre Bolívar permite encontrar rápidamente sinónimos e hiperónimos de los términos que el usuario introduce en su búsqueda. Una vez encontrados estos términos relacionados; no sólo se muestran los resultados de la búsqueda de los términos originales, sino también los resultados posibles de la búsqueda si se hubiera realizado con cada uno de los términos relacionados. La salida se encuentra ordenada en los distintos tipos de términos, para que resulte más claro para el usuario.

4.5.1.- La WordNet de "Luces de Bolívar en la Red"

Para el sitio de Luces de Bolívar en la Red, se buscaron los términos más importantes, se ordenaron y relacionaron. La WordNet creada no deriva de otra WordNet, como se suele hacer para ámbitos generales. Esta WordNet es exclusiva para el dominio de Bolívar. Esto se hizo con la idea de contar con definiciones y términos lo más ajustados posible al ámbito de Simón Bolívar. Los archivos lexicográficos que son usados para la creación de la WordNet fueron llenados únicamente con los términos obtenidos del sitio de Luces de Bolívar en la Red y con algunos

términos generales que se utilizan en búsquedas. Es importante notar que, para un sistema en producción, es conveniente que el trabajo de creación de estos archivos lexicográficos sea realizado por lingüistas y por especialistas en el área de estudio, quienes, seguramente, pueden reunir mejores criterios de clasificación taxonómica de la terminología que requieren las entradas al WordNet. Sin embargo, a efectos de probar la utilidad y factibilidad de una WordNet temática, que sirva para la realización de consultas semánticas, no se convocó la colaboración de estos profesionales para el presente trabajo.

Los términos utilizados para la creación del árbol semántico; es decir, la estructura de relaciones entre significados del sitio Luces de Bolívar en la Red se pueden apreciar en el ANEXO C.

Si se desea ampliar esta WordNet, se deben añadir nuevos términos a los archivos lexicográficos y estos deben ser compilados con el programa "grind"; incluido con las librerías que se copian al instalar la versión en inglés de WordNet. Un extracto del archivo lexicográfico noun.person (donde se deben almacenar los sustantivos referentes a personas) es:

```
{ poeta, escritor,@ (persona que escribe obras poéticas y está dotada de las facultades necesarias para componerlas) }
{ politico, noun.Tops:persona,@ (persona que escribe sobre historia) }
{ teniente_coronel, militar,@ (Jefe de graduación inmediatamente superior al comandante e inferior al coronel) }
{ ambrosio_plaza, coronel,@ (Oficial del Ejército de Venezuela en la Guerra de Independencia, 1791-1821) }
{ alejandro_de_humboldt, humboldt, noun.Tops:persona,@ (naturalista, geólogo, mineralogista, astrónomo, explorador, sismólogo, vulcanista, demógrafo alemán) }
{ alexandre_dehollain, amigo,@ (amigo de Bolívar al que pidió dinero en Francia para viajar a Venezuela) }
```

En el extracto de archivo lexicográfico se observa, entre otras cosas, que:

- No se aceptan espacios en el nombre de los términos: Los espacios son sustituidos por guión bajo (_)
- Cada línea o conjunto delimitado entre { y } especifica un significado o "synset"
- La palabra anterior al símbolo @ (arroba) indica el hiperónimo del synset. Por ejemplo: un hiperónimo de "teniente_coronel" es "militar"
- El significado se indica, en la parte final del synset, utilizando paréntesis
- Los términos sinónimos se separan por coma (,): Por ejemplo "alejandro_de_humboldt" será equivalente a mencionar "humboldt"

Para ampliar la WordNet, se deben agregar definiciones de este tipo a los archivos lexicográficos. En la WordNet existen 44 archivos lexicográficos, divididos en nombres, verbos, adjetivos y adverbios:

Nombres		Verbos	Adjetivos	Adverbios
noun.act noun.animal noun.artifact noun.attribute noun.body noun.cognition noun.communication noun.event noun.feeling noun.food noun.group noun.location noun.motive	noun.object noun.person noun.phenomenon noun.plant noun.possession noun.process noun.quantity noun.relation noun.shape noun.state noun.substance noun.time noun.Tops	verb.body verb.change verb.cognition verb.communication verb.competition verb.consumption verb.contact verb.creation verb.emotion verb.perception verb.possession verb.social verb.stative verb.weather	adj.all adj.pert adj.ppl	adv.all

La compilación de estos archivos es lo que genera la WordNet. Los significados (synsets) deben ser definidos en el archivo lexicográfico que le corresponda. Para mayor información sobre la estructura de estos archivos y sus reglas, dirigirse a la documentación respectiva **[16]**. En el ANEXO B se puede apreciar el archivo lexicográfico noun.communication desarrollado para la WordNet de Bolívar.

4.5.2.- Módulo de interacción en C con la WordNet

WordNet provee una biblioteca de funciones en C para su acceso e interrogación. Si bien este código provee todas la funciones que se requieren para interactuar con la WordNet, se prefirió crear una interfaz más amigable y encapsular la biblioteca WordNet en una dll (Dynamic-link library) para que pudiera ser tratada como un componente por Alejandría en el sistema operativo Windows.

Las funciones “publicadas” por la dll son sólo dos:

- int iniciarWordNet()
- buscar (int tipo, char * palabra)

iniciarWordNet permite establecer la comunicación con la WordNet a la que tenga acceso el sistema operativo donde se ejecute Alejandría o el programa que utilice la dll.

buscar permite interrogar a la WordNet, una vez realizada la conexión, sobre los sinónimos, hiperónimos, hipónimos, homónimos y merónimos de una palabra o término. A través del parámetro *tipo* se indica qué se desea consultar.

El software WordNet que provee la Universidad de Princeton se rige por una

licencia de software libre propia, que permite usar y realizar modificaciones al código provisto siempre y cuando se reconozca la autoría de la Universidad de Princeton. El software de la WordNet fue encapsulado en una dll y se le creó una pequeña interfaz que se acoge a la licencia GPL2.

En el ANEXO D se puede apreciar el módulo principal de la dll desarrollada.

4.6- Ejemplo: ¿Quién ha escrito sobre Manuelita?

En el punto 1.4 del presente documento se mostró el proceso de realización de una búsqueda en Alejandría y el resultado arrojado por la aplicación. La consulta fue “Manuelita”. Se quería en realidad interrogar sobre “¿Quién ha escrito sobre Manuelita?”. A continuación, se muestra una consulta realizada a través de la nueva interfaz de Consulta Semántica y se compara el resultado obtenido con el obtenido de la búsqueda sintáctica que se utiliza hasta la fecha en Alejandría.

La realización de la consulta a través de la nueva interfaz, se muestra a continuación (Figura 8):

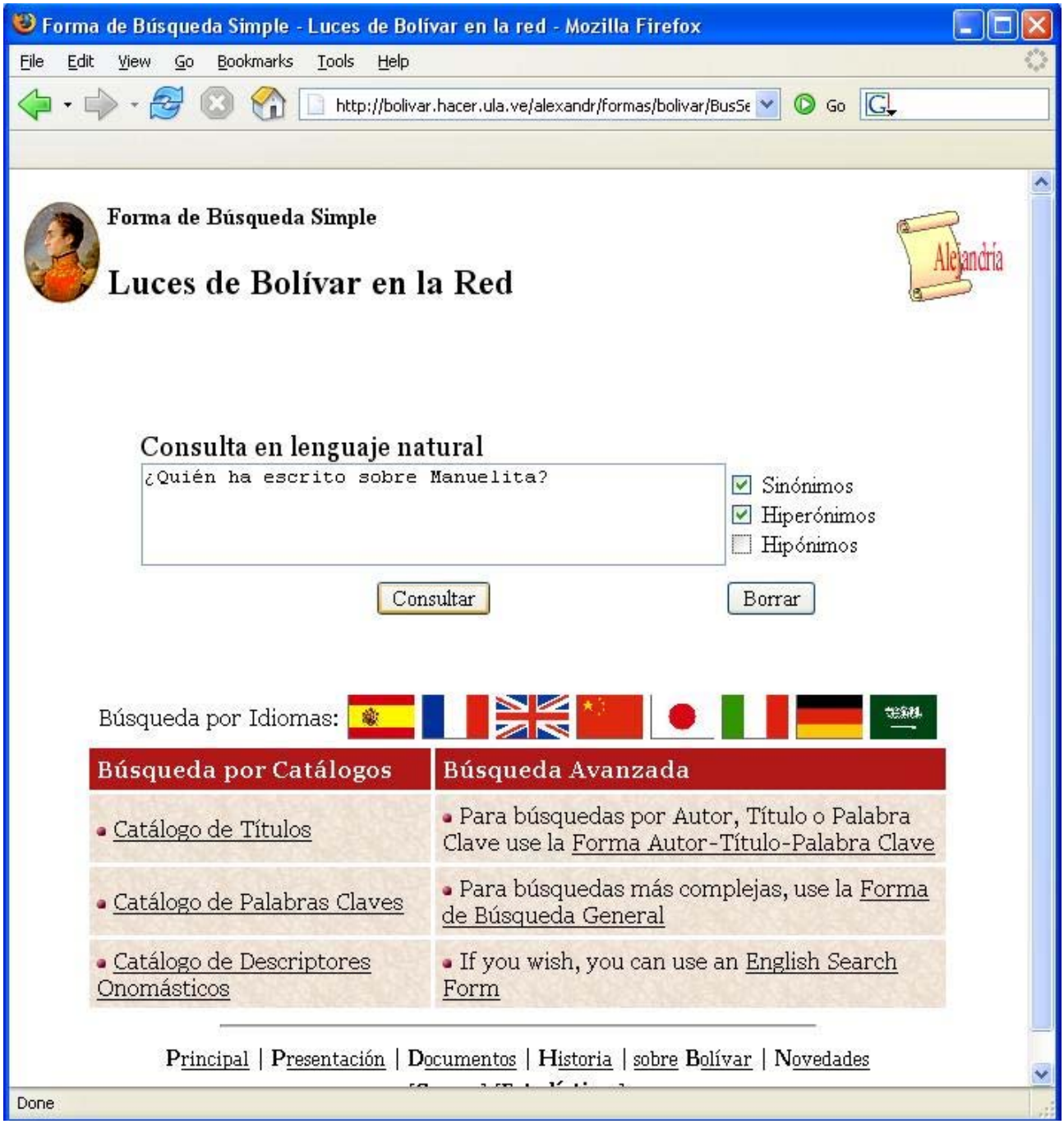


Figura 8: Interfaz de consulta semántica con la consulta "¿Quién ha escrito sobre Manuelita?

Como se puede apreciar, el usuario introduce la consulta en lenguaje natural, marca los tipos de términos relacionados que tomará en cuenta la consulta (Sinónimos, Hiperónimos o Hipónimos) y presiona el botón de Consultar. De esta forma, se realiza una búsqueda en lenguaje natural a través de la nueva interfaz.

Al presionar el botón de Consultar, en la interfaz de Consulta Semántica, el sistema identifica, a través de las gramáticas programadas con Prolog, que la consulta introducida por el usuario corresponde al patrón 1 de preguntas, donde se conoce que el usuario desea saber los autores que han escrito sobre cierto tema. Los extractos del archivo de consultas que realizan esta labor de identificación son:

```
consulta(1,X) --> quien, verbo_estudiar, tema(X).
…
verbo_estudiar --> [ha, escrito].
…
tema(T, [sobre|T], []).
```

El sistema comienza el procesamiento de la consulta de tipo 1; buscando los sinónimos e hiperónimos (según lo introducido en la interfaz) del tema: Manuelita. Según la WordNet creada (ver ANEXO C), se consigue como sinónimo de Manuelita el término "manuela_saenz" y como hiperónimo a "amante". Según se programó, el sistema sólo busca los hiperónimos directos; de lo contrario, mostraría también como hiperónimo a "persona". Esto se hizo porque se observó que los hiperónimos de niveles superiores no tienen mucha relación o pierden exactitud, con respecto al término buscado. En las pruebas realizadas, los hiperónimos directos resultaron apropiados, mientras que los de niveles superiores hacían involucrar términos muy generales; que resultaban poco eficaces para las búsquedas.

Con los sinónimos e hiperónimos, el módulo de procesamiento de las consultas genera un archivo HTML en memoria, con las búsquedas que se realizarán. El archivo generado y las consultas resaltadas pueden verse a continuación:

```
<html>
<body>
<SFA Enca;>
<H2>Consulta Semántica</H2>
<Table border=1>
<tr><td valign="_top"><h2>TÉRMINO ORIGINAL</h2>
<SFA CAle palabra=manuelita&pie=0&recuperar=5;>
</td></tr>
<tr><td valign="_top"><h2>SINÓNIMOS</h2>
<SFA CAle palabra=manuela+saenz&pie=0&recuperar=5;>
</td></tr>
<tr><td valign="_top"><h2>HIPERÓNIMOS</h2>
<SFA CAle palabra=amante&pie=0&recuperar=5;>
</td></tr>
</table>
<SFA Pie;>
</body>
</html>
```

Este archivo es procesado por Alejandría, que emite el resultado que se puede apreciar a continuación:

Figura 9: Resultado de la consulta semántica "¿Quien ha escrito sobre Manuelita?"

4.7.- Comparación y análisis de resultados

Como se aprecia en las salidas sintáctica (Figura 3) y semántica (Figura 9), se obtienen los mismos registros para el término original:

- Gómez, Andreína (Manuelita Sáenz: El enigma de una mujer 200 años después)
- Lovera De-Sola, R.J. (Manuelita Sáenz: una pasión desbocada)

Este era un resultado esperado, pues es uno de los resultados previstos en la salida para la consulta basada sólo en sintaxis. Pero en la salida semántica se obtienen además, para el caso de los sinónimos, 21 registros; esto no quiere decir que existan 21 autores más, sino que hay 21 documentos más que tratan sobre Manuelita; pero que fueron conseguidos a través del sinónimo "Manuela Saenz". Como se puede apreciar entre los primeros registros, algunos son cartas de Simón Bolívar y otros son documentos que no tienen autor, por lo que aparece sólo el título del documento. Entre los autores que se encuentran en registros posteriores (no mostrados en la captura de pantalla de la Figura 10), además de Simón Bolívar, están: "Nazoa, Aquiles", "Salcedo Bastardo, José Luis" y "Pérez Vila, Manuel".

En cuanto a los hiperónimos, la aplicación consiguió el hiperónimo "amante", pero no consiguió ningún texto que tuviese este término. Esto evidencia una falla en la realización de la WordNet: Se introdujo el término "amante" para referirse a un conjunto de personas, pero en la base de datos Alejandría, no se catalogó usando este término. Esta falta de sincronización hace que no se consigan más registros que pudieran ser relevantes para cierto usuario; ya que entre la información que se encuentra almacenada, sí existen documentos sobre otras amantes de Bolívar y por tanto, otros autores; si este fuese el tópico de interés.

Para la salida de la consulta semántica; se decidió para este tipo de pregunta, que se muestre también el título del documento; y no sólo el autor, debido a que existen documentos que no presentan autor.

Como se pudo apreciar, con un mínimo de procesamiento adicional, se lograron obtener los nombres de más autores que habían escrito sobre el tema. La introducción de la consulta a través de lenguaje natural, permitió interpretar mejor la búsqueda del usuario y la WordNet temática permitió conseguir más información para la consulta que se está respondiendo: Las herramientas semánticas resultaron útiles.

Del uso de ambas interfaces y de las salidas obtenidas, se puede inferir que:

- Las consultas en lenguaje natural pueden especificar mejor la pregunta; permitiendo ofrecer una respuesta más exacta

- La salida OWL de Alejandría sigue o cumple con la ontología creada; es decir, los metadatos que la conforman están documentados y disponibles públicamente para ser consultados. Con lo que se facilita su interpretación por usuarios externos o por un producto informático.
- Además de la búsqueda del concepto introducido, se enriquece la búsqueda con los sinónimos, hiperónimos y homónimos
- En la arquitectura planteada, el catalogador puede indicar nuevos sinónimos, hiperónimos e hipónimos a través de la WordNet temática. Esta WordNet temática debe ser construida, es única por tema y se puede ir mejorando progresivamente.
- La red semántica creada a través de la WordNet y los términos de los documentos puede tener varias visiones o interpretaciones subjetivas de relaciones, según se conceptualice. Relaciones de sinonimia, hiperonimia y homonimia que se pueden ir mejorando según se requiera.
- Una apropiada utilización de la arquitectura, permite obtener un mayor número de registros que tratan sobre el tema y permite conocer qué otros términos y registros están relacionados
- El hacer la WordNet sobre un área acotado del conocimiento, permite ser más exacto en los términos, definiciones y por tanto en los resultados. Se reducen los falsos positivos en los resultados de las búsquedas.

La desventaja principal del sistema semántico es que la búsqueda requiere más recursos. Puede ser más lenta, pues se realizan tantas búsquedas como elementos nuevos (sinónimos, hiperónimos, hipónimos) se consigan.

CONCLUSIONES

El presente trabajo, permitió comprobar que las tecnologías Web semánticas ofrecen una alternativa efectiva para estandarizar el intercambio y publicación de información, su catalogación, búsqueda y recuperación.

Las tres tecnologías utilizadas fueron: OWL, WordNet y Prolog. Estas resultaron útiles en la corrección de los errores de pérdida de semántica que típicamente ocurren al intercambiar información en lenguaje natural a través de medios electrónicos: OWL permite, definir y estructurar la información compartida en un formato compresible para el computador; WordNet permite relacionar los conceptos de los que se habla; y las gramáticas en Prolog permiten rescatar la semántica del lenguaje natural.

El trabajo de catalogar bien la información, a través de un lenguaje como OWL; muestra claramente su utilidad al momento de compartir y buscar información, pues la define y estructura, dando mayores posibilidades que los métodos estadísticos que se utilizan actualmente. Adicionalmente, permite la comparación con otras fuentes de información, de una forma estándar y utilizable por mecanismos electrónicos como el computador.

A pesar de que no se utilizó todo el potencial de WordNet, resultó un gran hallazgo y permitió mejorar los resultados de las consultas de usuarios; al servir como diccionario electrónico de sinónimos, hiperónimos, holónimos, merónimos e hipónimos. WordNet dio la posibilidad de ir más allá de una buena catalogación, al servir como instrumento activo para mejorar las búsquedas. El poder construir WordNets temáticas, para lenguajes locales o de nicho, aporta también gran valor, pues le otorga mayor exactitud a los términos relacionados al restringirlos a un ámbito acotado.

El trabajo permitió mostrar que el lenguaje natural y el tratamiento de la información a través de este, utilizando gramáticas específicas codificadas en Prolog, es, además de posible, beneficioso: Se rescata la semántica de lo que el usuario desea expresar, la cual se pierde (en parte) al codificar los mensajes. También es claro que por razones prácticas, no es conveniente utilizar gramáticas formales de grandes segmentos de un lenguaje natural, sino restringirse a ámbitos del lenguaje acotados, que bien pueden definirse con esas gramáticas específicas. Mientras más específica sea la gramática, será más fácil evaluar, corregir y mantener el sistema de consulta.

Los resultados obtenidos incumben a, prácticamente, cualquier aplicación de manejo de información: No sólo a Alejandría.

RECOMENDACIONES

Tal y como se planteó desde un inicio, el trabajo realizado presenta un experimento con tecnologías Web semánticas para abordar algunos problemas de manejo de información que siguen sin ser solucionados de forma idónea.

Tratándose de un experimento acotado en su alcance y sus soluciones, resulta claro que se debe seguir trabajando en esta área; ya que se obtuvieron resultados positivos que muestran que las tecnologías semánticas constituyen una mejor opción para la solución de ciertos problemas de gestión de información, comparadas con el uso de otros mecanismos de inteligencia artificial o estadísticos; que no abordan uno de los objetivos primordiales: Facilitar a los computadores la "comprensión" o procesamiento de la información compartida.

La WordNet es una herramienta muy valiosa, ya que relaciona, clasifica y estructura los metadatos en un dominio. Sin embargo, sólo se pudieron aprovechar una parte de sus bondades; porque, por razones de licenciamiento, se debió usar la versión en inglés. Se recomienda convertir la WordNet en español (EuroWordNet) en libre o la construcción de una nueva WordNet en español amparada bajo alguna licencia de software libre. Esto permitiría la utilización de las estructuras lingüísticas propias de nuestro idioma y de caracteres como la ñ y las tildes. Se hace un gran daño al restringir el acceso a una herramienta con este potencial.

Con una WordNet libre sobre el español, las aplicaciones de gestión de información como Alejandría, podrían evaluar una eventual conveniencia, de derivar sus WordNets temáticas (descriptores), de la WordNet en español. Esto ahorraría trabajo y añadiría estandarización; al convertirse la WordNet en español en una ontología global para nuestro idioma.

Con una gran parte del lenguaje estandarizado, restaría por ser compartido de forma estándar, a través de OWL o como WordNets especializadas, el argot o lenguaje específico de los dominios en los cuales se enfocan algunas aplicaciones.

Este trabajo también permitió ver, que las consultas en lenguaje natural, deben ser promovidas, sobre todo en aquellas condiciones donde resultan más efectivas que las consultas comunes, ya que hay un contenido semántico que se pierde al codificar las consultas. Se recomienda, por efectos prácticos, la utilización de lenguaje natural en ambientes donde existe un lenguaje controlado; que no es el caso, por ejemplo, de los buscadores de información en Internet.

En Alejandría, pudiera parecer innecesario el uso de lenguaje natural, ya que generalmente se

diseñan las aplicaciones con la idea de tener campos claves por los cuales buscar; sin embargo, la utilización de la WordNet como repositorio de descriptores o tal vez el enriquecimiento de la funcionalidad de descriptores que existe actualmente con sinónimos, hiperónimos y homónimos; fortalecería e incrementaría la calidad de las búsquedas.

Un área donde el W3C está comenzando a dar una respuesta firme es en las tecnologías de recuperación de datos en RDF (y OWL). Con la aparición del SPARQL (SPARQL Protocol and RDF Query Language) y el soporte cada vez mejor que tienen los manejadores de base de datos para albergar tipos de datos XML, se puede comenzar a pensar en almacenar los datos en OWL. Sin embargo, aún pareciera que estas tecnologías se encuentran en un nivel experimental; pues los pasos para consultar un campo siguen siendo: Recuperar el campo XML a memoria, estructurar estos datos en memoria según una definición RDF u OWL y realizar el query o consulta sobre uno de los campos que almacena el texto XML.

ANEXOS

ANEXO A

Archivo Prolog para la identificación de las consultas: *consultas.pl*

```
% consulta.pl
%
% Authors:        Jacinto Dávila and Icaro Alzuru
% E-mail:         jacinto@ula.ve, ialzuru@hotmail.com
% Copyright (C): 2006, Jacinto Dávila and Icaro Alzuru, Venezuela
%
% This program is free software; you can redistribute it and/or modify it under
% the terms of the GNU General Public License as published by the Free Software
% Foundation; either version 2 of the License, or (at option) any later version.
%
% This program is distributed in the hope that it will be useful, but WITHOUT
% ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
% FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License along with
% this library; if not, write to the Free Software Foundation, Inc., 59 Temple
% Place, Suite 330, Boston, MA  02111-1307 USA
%
% You can also find a copy of the GPL at http://www.gnu.org/copyleft/gpl.html
%
%%%%%%%
% Programa para identificar las consultas de un usuario en el proyecto de
% Alejandría inteligente.


%consulta 1:
consulta(1,X) --> quien, verbo_estudiar, tema(X).
consulta(1,X) --> alguien, verbo_estudiar, tema(X).
consulta(1,X) --> verbo_mostrar, nombre_autor, que, verbo_estudiar, tema(X).
consulta(1,X) --> nombre_autor, de, nombre_escritos, tema(X).

%consulta 2:
consulta(2,X) --> donde, consigo_inf, sobre, articulo, tema(X).
consulta(2,X) --> donde, consigo_inf, sobre, tema(X).
consulta(2,X) --> articulos_interrogativos, monografia, tratan, sobre, articulo,
tema(X).
consulta(2,X) --> articulos_interrogativos, monografia, tratan, sobre, tema(X).

%consulta 3:
consulta(3,X) --> como, hago, articulo, tema(X).
consulta(3,X) --> como, hago, tema(X).

%consulta 4:
consulta(4,X) --> que, es, articulo, tema(X).
consulta(4,X) --> que, es, tema(X).

%Consulta 5: Es la misma Consulta 1
consulta(1,X) --> a, quien, verbo_consultar, sobre, tema(X).

%Consulta 6:
consulta(6,X) --> articulo, tema(X), es, parte, de, que.
consulta(6,X) --> de, que, es, parte, articulo, tema(X).
consulta(6,X) --> a, que, pertenece, articulo, tema(X).

a --> [a].

alguien --> [alguien].
alguien --> [alguno].
```

```
alguien --> [alguna].

articulo --> [el].
articulo --> [la].
articulo --> [los].
articulo --> [las].
articulo --> [un].
articulo --> [unos].
articulo --> [una].
articulo --> [unas].
articulo --> [].

articulos_interrogativos --> [que].
articulos_interrogativos --> [en, que].

como --> [como].

consigo_inf --> [consigo, informacion].
consigo_inf --> [se, consigue, informacion].
consigo_inf --> [consigo, algo].
consigo_inf --> [encuentro, informacion].
consigo_inf --> [se, encuentra, informacion].
consigo_inf --> [encuentro, algo].
consigo_inf --> [habla].
consigo_inf --> [hablan].
consigo_inf --> [se, habla].

de --> [de].
de --> [de, los].

donde --> [donde].
donde --> [en, donde].
donde --> articulos_interrogativos, monografia.

es --> [es].
es --> [significa].
es --> [consiste].
es --> [son].
es --> [forman].
es --> [forma].

hago --> [hago].
hago --> [se, hace].
hago --> [fabrico].
hago --> [se, fabrica].
hago --> [construyo].
hago --> [se, construye].

monografia --> [monografia].
monografia --> [monografias].
monografia --> [libro].
monografia --> [libros].
monografia --> [texto].
monografia --> [textos].
monografia --> [documento].
monografia --> [documentos].
monografia --> [escrito].
monografia --> [escritos].
```

```
nombre_autor --> [autores].
nombre_autor --> [escritores].
nombre_autor --> [estudiosos].
nombre_autor --> [investigadores].
nombre_autor --> articulo, nombre_autor.

nombre_escritos --> [textos].
nombre_escritos --> [libros].
nombre_escritos --> [escritos].
nombre_escritos --> [monografias].

parte --> [una, parte].
parte --> [parte].
pertenece --> [pertenece].

que --> [que].
que --> [en, que].
quien --> [quien].
quien --> [quienes].

sobre --> [sobre].
sobre --> [a, cerca, del].
sobre --> [a, cerca, de].
sobre --> [de].
sobre --> [del].

tratan --> [trata].
tratan --> [tratan].
tratan --> [habla].
tratan --> [hablan].
tratan --> [hay].

verbo_consultar --> [consulto].

verbo_estudiar --> [ha, escrito].
verbo_estudiar --> [han, escrito].
verbo_estudiar --> [ha, hablado].
verbo_estudiar --> [han, hablado].
verbo_estudiar --> [ha, estudiado].
verbo_estudiar --> [han, estudiado].
verbo_estudiar --> [ha, trabajado].
verbo_estudiar --> [han, trabajado].

verbo_mostrar -->  [lista].
verbo_mostrar -->  [muestra].
verbo_mostrar -->  [enumera].
verbo_mostrar -->  [despliega].
verbo_mostrar -->  [busca].
verbo_mostrar -->  [di].
verbo_mostrar -->  [indica].
verbo_mostrar -->  [presenta].
verbo_mostrar -->  [revela].
verbo_mostrar -->  [especifica].
verbo_mostrar -->  [menciona].
verbo_mostrar -->  [expon].            % hay problemas con el acento.
verbo_mostrar -->  [nombra].
verbo_mostrar -->  [listame].          % hay problemas con el acento.
```

```
tema(T, [algo, de|T], []).
tema(T, [algo, sobre|T], []).
tema(T, [acerca,de|T], []).
tema(T, [sobre|T], []).
tema(X, X, []).
tema([], X, X).
```

ANEXO B

Archivo lexicográfico: *noun.communication*

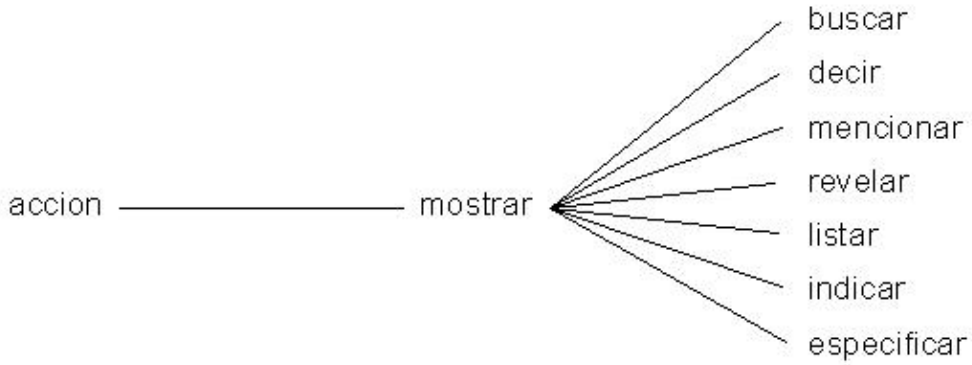
```
(noun.communication)
(
Author:        Icaro Alzuru
E-mail:        ialzuru@hotmail.com
Copyright (C): 2006, Icaro Alzuru, Venezuela

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 2 of the License, or (at your option) any later
version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.  See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with
this library; if not, write to the Free Software Foundation, Inc., 59 Temple
Place, Suite 330, Boston, MA 02111-1307 USA

You can also find a copy of the GPL at http://www.gnu.org/copyleft/gpl.html
)
{ acuarela, tecnica_artistica,@ pintura,@ (pintura sobre papel o cartulina con
colores diluidos en agua) }
{ alocucion, noun.Tops:obra,@ (Discurso o razonamiento breve por lo común y
dirigido por un superior a sus inferiores, secuaces o súbditos) }
{ biografia, escrito,@ (relato que puede ser didactico o expositivo de la
historia de la vida de una persona) }
{ carboncillo, tecnica_artistica,@ dibujo,@ (Técnica pictórica donde el dibujo
se crea con un palillo carbonizado de brezo, sauce u otra madera ligera) }
{ carta, escrito,@ (papel escrito, y ordinariamente cerrado, que una persona
envía a otra para comunicarse con ella) }
{ catalogo, escrito,@ (relación ordenada en la que se incluyen o describen de
forma individual libros, documentos, personas, objetos, etc., que están
relacionados entre sí) }
{ coleccion, noun.Tops:obra,@ (conjunto ordenado de cosas, por lo común de una
misma clase y reunidas por su especial interés o valor) }
{ convocatoria, escrito,@ (anuncio o escrito con que se convoca, citación) }
{ cuadro, noun.Tops:obra,@ (lienzo, lámina, papel, etc., de una pintura, un
grabado, un dibujo o similar) }
{ decreto, escrito,@ (decisión tomada por la autoridad competente en materia de
su incumbencia, y que se hace pública en las formas prescritas) }
{ dialogo, escrito,@ (plática entre dos o más personas, que alternativamente
manifiestan sus ideas o afectos) }
{ diario, escrito,@ (cuaderno o libro en que se recogen acontecimientos y
pensamientos día a día) }
{ dibujo, noun.Tops:obra,@ (técnica gráfica basada en el uso de la línea. Se
realiza normalmente sobre papel, cartón, etc. Puede emplear el color o
prescindir de él) }
{ diccionario, libro,@ (libro en el que se recogen y explican de forma ordenada
voces de una o más lenguas, de una ciencia o de una materia determinada) }
{ discurso, alocucion,@ (razonamiento o exposición sobre algún tema que se lee o
pronuncia en público) }
{ escrito, noun.Tops:obra,@ (carta, documento o cualquier papel manuscrito,
mecanografiado o impreso) }
{ grabado, tecnica_artistica,@ (arte y procedimiento de grabar una imagen sobre
una superficie) }
{ ideario, escrito,@ (repertorio de las principales ideas de un autor, de una
escuela o de una colectividad) }
```

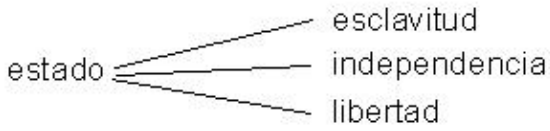
```
{ lapiz, tecnica_artistica,@ (nombre genérico de varias sustancias minerales que
sirven para dibujar) }
{ libro, escrito,@ (obra científica, literaria o de cualquier otra índole con
extensión suficiente para formar volumen, que puede aparecer impresa o en otro
soporte) }
{ litografia, tecnica_artistica,@ (técnica de reproducir, mediante impresión, lo
dibujado o grabado previamente en una piedra caliza) }
{ manifiesto, escrito,@ (escrito en que se hace pública declaración de doctrinas
o propósitos de interés general) }
{ memoria, escrito,@ (relación escrita de actividades) }
{ mensaje, escrito,@ (recado de palabra o escrito que una persona envía a otra)
}
{ miniatura, tecnica_artistica,@ (pintura de pequeño tamaño, hecha con mucho
detalle sobre una superficie delicada, en especial la que ilustraban manuscritos
antiguos) }
{ monografia, escrito,@ (estudio o investigación sobre un tema particular) }
{ mosaico, tecnica_artistica,@ (técnica artística de decoración que se forma
pegando sobre un fondo de cemento pequeñas piezas de piedra, vidrio o cerámica
de diversos colores para formar dibujos) }
{ oleo, tecnica_artistica,@ (pintura que se obtiene disolviendo ciertos
pigmentos en una solución aceitosa) }
{ palabras, alocucion,@ (lo que dice alguien o está escrito en algún texto. Más
en plural) }
{ pensamiento, escrito,@ (conjunto de ideas propias de una persona o
colectividad) }
{ pintura, noun.Tops:obra,@ (tabla, lámina o lienzo en que está pintado algo) }
{ poema, escrito,@ (obra en verso, o perteneciente por su género a la esfera de
la poesía aunque esté escrita en prosa) }
{ proclama, alocucion,@ (alocución política o militar, de viva voz o por
escrito) }
{ retrato, noun.Tops:obra,@ (pintura, dibujo, fotografía, etc., que representa
alguna persona o cosa) }
{ tecnica_artistica, noun.Tops:tecnica,@ (conjunto de procedimientos y recursos
para crear arte) }
{ testamento, escrito,@ (declaración que de su última voluntad hace una persona,
disponiendo de bienes y de asuntos que le atañen para después de su muerte) }

{ angostura, discurso_de_angostura, discurso,@ (Discurso pronunciado por Simón
Bolívar el 15 de febrero de 1819, en la provincia de Guayana, con motivo de la
instalación del segundo Congreso Constituyente de la República de Venezuela en
San Tomé de Angostura (hoy Ciudad Bolívar)) }
{ cartagena, manifiesto,@ (Documento político escrito por Simón Bolívar el 15 de
diciembre en la ciudad de Cartagena de Indias (Colombia)) }
{ carupano, manifiesto,@ (Carúpano, septiembre 7 de 1814) }
{ congreso_admirable, mensaje,@ (Bogotá, enero 20 de 1830.) }
{ congreso_constituyente_de_bolivia, mensaje,@ (25 de mayo de 1826) }
{ congreso_de_cucuta, discurso,@ (3 de octubre de 1821) }
{ congreso_de_peru, palabras,@ (13 de septiembre de 1823) }
{ guerra_a_muerte, decreto,@ (Escrito realizado en el Cuartel General de
Trujillo, 15 de junio de 1813. Fue la respuesta de Bolívar ante los numerosos
crímenes perpetrados por Domingo de jefes realistas luego de la caída de la
Primera República) }
{ jamaica, carta,@ (Carta de Bolívar escrita en Kingston, setiembre 6 de 1815) }
{ libertad_de_los_esclavos, proclama,@ (Discurso dado en el Cuartel General de
Ocumare, 6 de julio de 1816) }
{ naciones_del_mundo, naciones, manifiesto,@ (20 de Septiembre. Manifiesto a las
Naciones del Mundo sobre los acontecimientos de Venezuela en los años 1812 y
1813) }
```

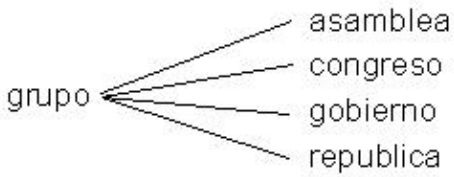
{ sociedad_patriotica, discurso,@ (Discurso dado del 3 al 4 de julio de 1811 para la Sociedad Patriótica (organización revolucionaria proindependentista)) }
{ ultima_proclama, proclama,@ (Proclama antes de su muerte: Hacienda de San Pedro, en Santa Marta, a 10 de diciembre de 1830) }

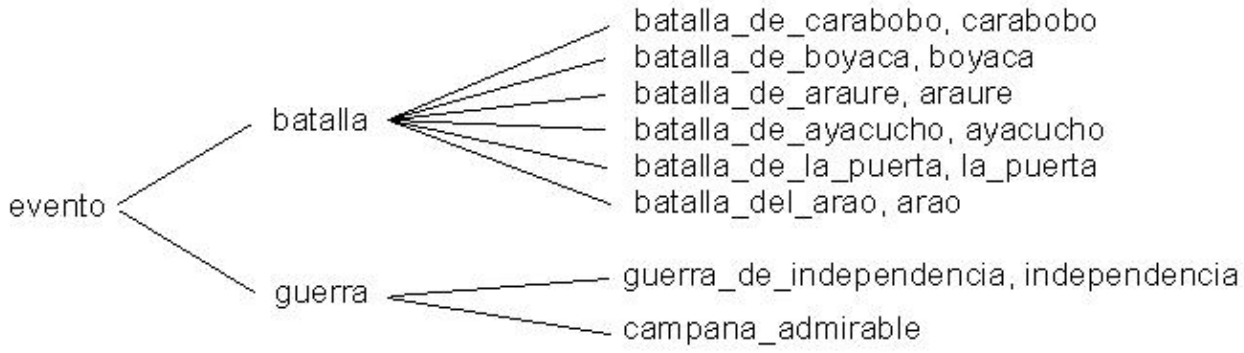
ANEXO C

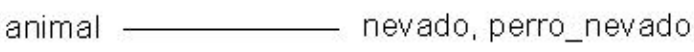
Estructura de relaciones entre significados y términos utilizados para la creación de la WordNet del sitio Web Luces de Bolívar en la Red

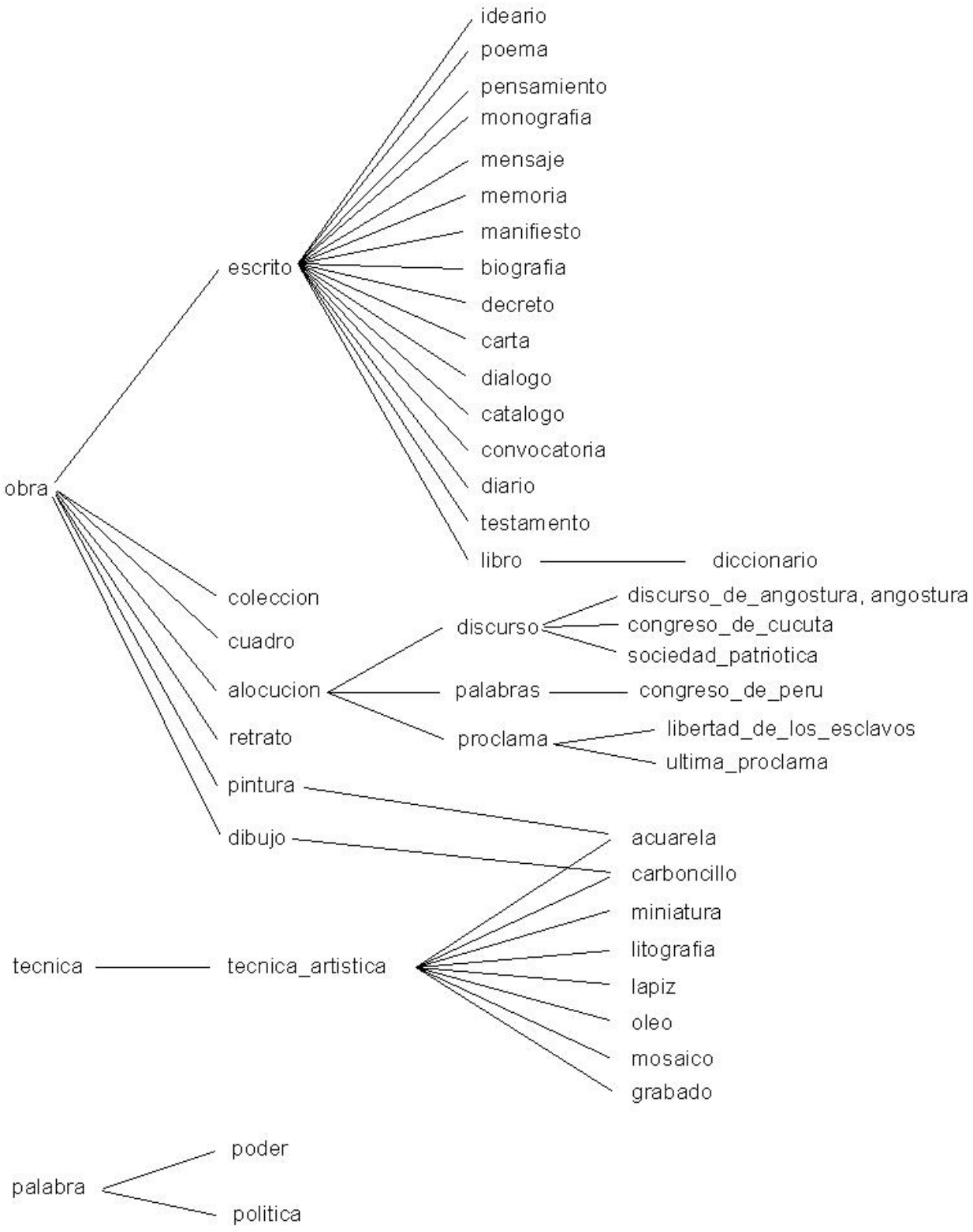

accion
mostrar
buscar
decir
mencionar
revelar
listar
indicar
especificar

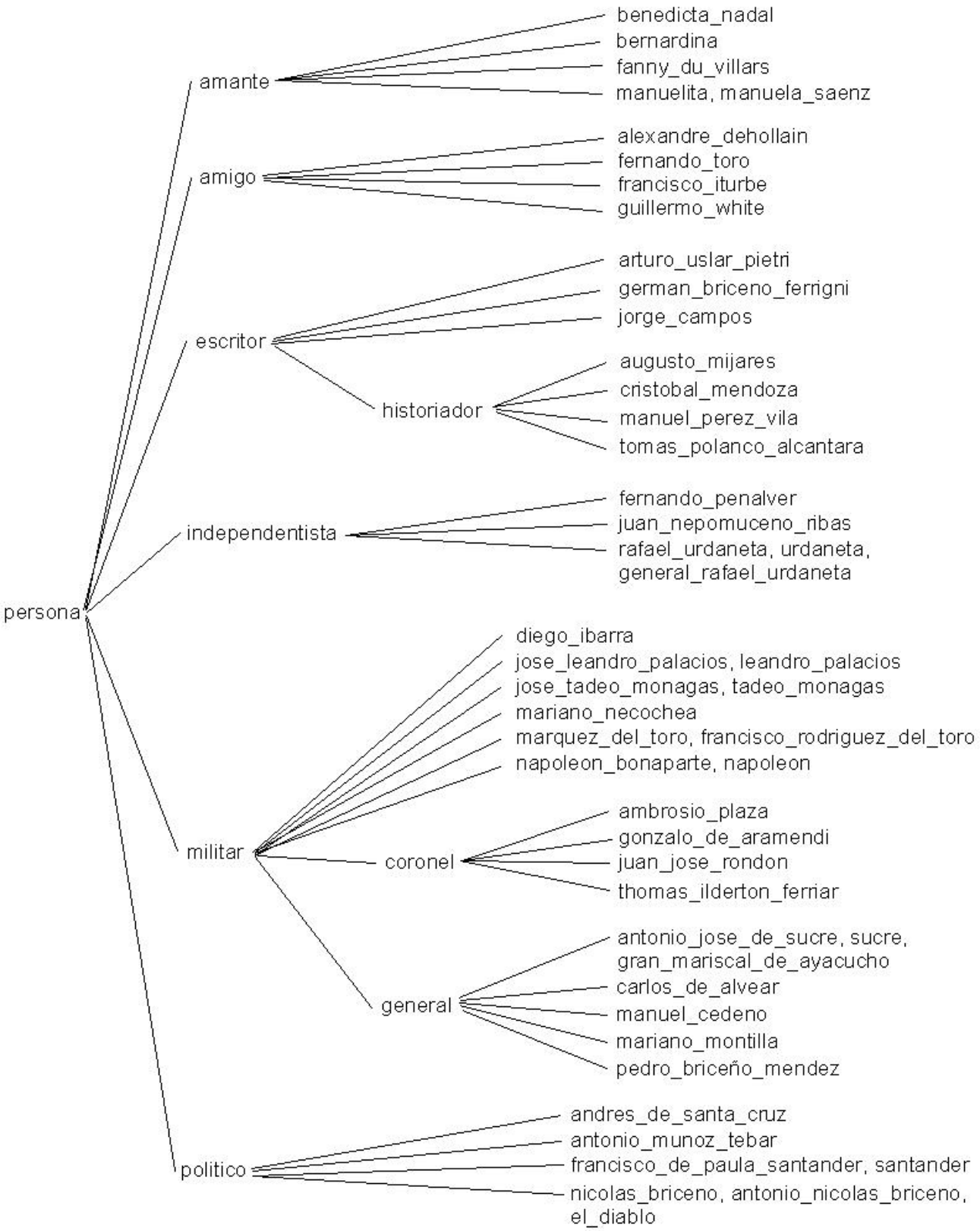

estado
esclavitud
independencia
libertad


grupo
asamblea
congreso
gobierno
republica


evento
batalla
batalla_de_carabobo, carabobo
batalla_de_boyaca, boyaca
batalla_de_araure, araure
batalla_de_ayacucho, ayacucho
batalla_de_la_puerta, la_puerta
batalla_del_arao, arao
guerra
guerra_de_independencia, independencia
campana_admirable


animal
nevado, perro_nevado


obra
escrito
ideario
poema
pensamiento
monografia
mensaje
memoria
manifiesto
biografia
decreto
carta
dialogo
catalogo
convocatoria
diario
testamento
libro
diccionario
coleccion
cuadro
alocucion
discurso
discurso_de_angostura, angostura
congreso_de_cucuta
sociedad_patriotica
palabras
congreso_de_peru
proclama
libertad_de_los_esclavos
ultima_proclama
retrato
pintura
dibujo
tecnica
tecnica_artistica
acuarela
carboncillo
miniatura
litografia
lapiz
oleo
mosaico
grabado
palabra
poder
politica


persona
amante
benedicta_nadal
bernardina
fanny_du_villars
manuelita, manuela_saenz
amigo
alexandre_dehollain
fernando_toro
francisco_iturbe
guillermo_white
escritor
arturo_uslar_pietri
german_briceno_ferrigni
jorge_campos
historiador
augusto_mijares
cristobal_mendoza
manuel_perez_vila
tomas_polanco_alcantara
independentista
fernando_penalver
juan_nepomuceno_ribas
rafael_urdaneta, urdaneta, general_rafael_urdaneta
militar
diego_ibarra
jose_leandro_palacios, leandro_palacios
jose_tadeo_monagas, tadeo_monagas
mariano_necochea
marquez_del_toro, francisco_rodriguez_del_toro
napoleon_bonaparte, napoleon
coronel
ambrosio_plaza
gonzalo_de_aramendi
juan_jose_rondon
thomas_ilderton_ferriar
general
antonio_jose_de_sucre, sucre, gran_mariscal_de_ayacucho
carlos_de_alvear
manuel_cedeno
mariano_montilla
pedro_briceño_mendez
politico
andres_de_santa_cruz
antonio_munoz_tebar
francisco_de_paula_santander, santander
nicolas_briceno, antonio_nicolas_briceno, el_diablo

ANEXO D

Módulo principal para la dll WordNet: *WordNet.cpp*

```
// wordnet.cpp: Defines the entry point for the DLL application.
/*
Author:        Icaro Alzuru
E-mail:        ialzuru@hotmail.com
Copyright (C): 2006, Icaro Alzuru, Venezuela

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 2 of the License, or (at your option) any later
version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.  See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with
this library; if not, write to the Free Software Foundation, Inc., 59 Temple
Place, Suite 330, Boston, MA 02111-1307 USA

You can also find a copy of the GPL at http://www.gnu.org/copyleft/gpl.html
*/

#include "stdafx.h"
#include "wordnet.h"

WORDNET_API int iniciarWordNet();
WORDNET_API SynsetPtr buscar (int tipo, char * palabra);


BOOL APIENTRY DllMain( HANDLE hModule, DWORD  ul_reason_for_call, LPVOID
lpReserved)
{
    switch (ul_reason_for_call)
      {
            case DLL_PROCESS_ATTACH:
            case DLL_THREAD_ATTACH:
            case DLL_THREAD_DETACH:
            case DLL_PROCESS_DETACH:
                  break;
    }
    return TRUE;
}

// This is the constructor of a class that has been exported.
// see wordnet.h for the class definition
CWordnet::CWordnet()
{
      return;
}

WORDNET_API int iniciarWordNet() {
   if (wninit())
      return 0;
   else
      return 1;
}

WORDNET_API SynsetPtr buscar (int tipo, char * palabra)
```

```
{
   switch (tipo)
   {
      case SINONIMO:
         return findtheinfo_ds(palabra,NOUN,SYNS,ALLSENSES);
      case HIPERONIMO:
         return findtheinfo_ds(palabra,NOUN,-HYPERPTR,ALLSENSES);
      case HIPONIMO:
         return findtheinfo_ds(palabra,NOUN,HYPOPTR,ALLSENSES);
      case MERONIMO:
         return findtheinfo_ds(palabra,NOUN,MERONYM,ALLSENSES);
      case HOLONIMO:
         return findtheinfo_ds(palabra,NOUN,HOLONYM,ALLSENSES);
   }
   return findtheinfo_ds(palabra,NOUN,SYNS,ALLSENSES);
                                          }
```

REFERENCIAS

[1] "**The global structure of an HTML document, HTML 4.01 Specification W3C Recommendation**", World Wide Web Consortium, 1999-12-24. http://www.w3.org/TR/html4/struct/global.html#h-7.4.4

[2] "**Namespaces in XML**", World Wide Web Consortium, 1999-01-14. http://www.w3.org/TR/REC-xml-names

[3] "**EuroWordNet, Building a multilingual database with WordNets for several European languages**", Department of Computational Linguistics, Universidad de Amsterdam, 2001-09. http://www.illc.uva.nl/EuroWordNet/

[4] "**RDF Primer**", World Wide Web Consortium, 2004-02-10. http://www.w3.org/TR/2004/REC-rdf-primer-20040210

[5] "**Alejandría, Sistema de gestión de información**", Mérida, Venezuela: Hacer Sistemas, 2006-07-17. http://www.alejandria.biz

[6] "**The Dublin Core Metadata Initiative**", Dublin, USA: 2006-07-17. http://www.dublincore.org/

[7] "**Google technology**", 2006-07-17. http://www.google.com/technology, http://www.google.com/technology/whyuse.html

[8] "**Luces de Bolívar en la Red**", 2006-07-17. http://www.bolivar.ula.ve

[9] "**MAchine-Readable Cataloging**", 2006-07-17. http://www.loc.gov/marc/marcspa.html

[10] "**Online Computer Library Center**", 2006-07-17. http://www.oclc.org

[11] "**Web Ontology Language**", 2006-06-29. http://www.w3.org/2004/OWL

[12] "**Prolog**", 2006-07-16. http://en.wikipedia.org/wiki/Prolog

[13] "**Protègè**", 2006-07-17. http://protege.stanford.edu/index.html

[14] "**Semantic Web**", World Wide Web Consortium, 2006-07-17. http://www.w3.org/2001/sw/

[15] "**Uniform Resource Identifier**", Wikipedia, 2006-04-29. http://es.wikipedia.org/wiki/URI

[16] "**WordNet Database**", USA: Universidad de Princeton, 2006-07-17. http://wordnet.princeton.edu/man/wndb.5WN.html

[17] "**Wikipedia – WordNet**", 2006-06-19. http://en.wikipedia.org/wiki/Wordnet

[18] "**WordNet: A lexical database for the English language**", USA: Universidad de Princeton, 2006-07-17. http://wordnet.princeton.edu

[19] "**eXtensible Markup Language**", 2006-05-22. http://www.w3.org/XML

[20] "**Metodología Alejandría para el Desarrollo de Sistemas de Teleinformación**", Versión 6.0.8. Mérida, Venezuela: Hacer Sistemas.

[21] Andy Carvin, "**Tim Berners-Lee: Weaving a Semantic Web**", EDC Center for Media & Community, 2005-02-01. http://www.digitaldivide.net/articles/view.php?ArticleID=20

[22] Chris Taylor, "**An Introduction to Metadata**", University of Queensland Library, 2003-07-29. http://www.library.uq.edu.au/iad/ctmeta4.html

[23] Christiane Fellbaum, "**WORDNET, An Electronic Lexical Database**", Cambridge, Massachusetts, USA: MIT Press, 1998.

[24] Dan Brickley y R.V.Guha, "**RDF Vocabulary Description Language 1.0: RDF Schema**", World Wide Web Consortium, 2004-02-10. http://www.w3.org/TR/rdf-schema

[25] Deborah L.McGuinness y Frank van Harmelen, "**OWL Web Ontology Language Overview**", 2004-02-10. http://www.w3.org/TR/2004/REC-owl-features-20040210/

[26] Hans Weigand, 1997.

[27] Matthew Horridge, "**A Practical Guide To Building OWL Ontologies With The Protègè-OWL Plugin**", Universidad de Manchester, 2004-06-13.

[28] Michael C.Daconta, Leo J.Obrst y Kevin T.Smith, "**The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management**", John Wiley & Sons, 2003.

[29] Natalya F.Noy y Deborah L.McGuinness, "**Ontology Development 101: A Guide to Creating Your First Ontology**", Stanford, USA: Universidad de Stanford.

[30] Pereira, Fernando C.N. y Stuart M.Shieber, "**Prolog and Natural-Language Analysis**", No. 10, Center of Study of Language and Information, 1987.

[31] Ronald Fagin, Joseph Y.Halpern, Yoram Moses y Moshe Y.Vardi, "**Reasoning About Knowledge**", MIT Press, 2003.

[32] Sergey Brin y Lawrence Page. "**The Anatomy of a Large-Scale Hypertextual Web Search Engine**", California, USA: Computer Science Department, Stanford University.

[33] Tim Berners-Lee, James Hendler y Ora Lassila, "**The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities**", Scientific American, 2001-05. http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21