UTILIZACIÓN DE WORDNET COMO ONTOLOGÍA PARA APOYAR LA RECUPERACIÓN DE DOCUMENTOS EN UN SISTEMA DE GESTIÓN DE INFORMACIÓN

MAURICIO ALBERTO MONSALVE RÍOS

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
ESCUELA DE INGENIERÍA
UNIVERSIDAD EAFIT
MEDELLÍN
2.004

UTILIZACIÓN DE WORDNET COMO ONTOLOGÍA PARA APOYAR LA RECUPERACIÓN DE DOCUMENTOS EN UN SISTEMA DE GESTIÓN DE INFORMACIÓN

MAURICIO ALBERTO MONSALVE RÍOS

Proyecto de grado para optar al título de
Ingeniero de Sistemas

Asesor
Juan Guillermo Lalinde Pulido
Ingeniero de Sistemas
Jefe del departamento de Ingeniería de Sistemas

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
ESCUELA DE INGENIERÍA
UNIVERSIDAD EAFIT
MEDELLÍN
2.004

Nota de Aceptación

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado

_____

Jurado

Medellín, 2 de Agosto de 2.004

A mi familia por estar siempre presentes en todos los momentos durante el transcurso de este proyecto, brindándome su amor y apoyo incondicional

AGRADECIMIENTOS

El autor expresa sus agradecimientos a:

Juan Guillermo Lalinde Pulido, Ingeniero de Sistemas, por brindarme el apoyo, su conocimiento y experiencia para el desarrollo del proyecto y también por la paciencia durante el desarrollo del mismo.

John Restrepo, por la oportunidad de realizar este trabajo con toda la información que tiene a su cargo en la Universidad Técnica de Delft, por sus valiosos aportes en el manejo del gestor de contenidos Ez Publish.

Laboratorio de Telemática Universidad EAFIT; por facilitarme el espacio y los recursos necesarios para el desarrollo del proyecto.

Y a todas aquellas personas que de una u otra forma acompañaron este proceso durante poco más de un año llenándome de alegría y entusiasmo para continuar adelante a pesar de las adversidades.

TABLA DE CONTENIDO

Pág.

TABLA DE ILUSTRACIONES

Pág.

1 INTRODUCCIÓN

Este trabajo de grado , titulado UTILIZACIÓN DE WORDNET COMO ONTOLOGÍA PARA APOYAR LA RECUPERACIÓN DE DOCUMENTOS EN UN SISTEMA DE GESTIÓN DE INFORMACIÓN, integra en la herramienta Ez Publish el recurso lingüístico WordNet para apoyar a los usuarios en los procesos de búsqueda de información.

En la primera parte se hace referencia a las motivaciones y las necesidades específicas a las que se desea darles solución con el desarrollo del proyecto.

En la segunda parte, podrá encontrar los conceptos teóricos que son definidos con la intención de ser el marco de referencia o la base conceptual; en esta sección se presentan las definiciones básicas, se enuncian temas relacionados directa o indirectamente con la investigación sin ser necesariamente parte de ésta, también se presentan las distintas tecnologías involucradas en el proyecto y, como tema central de investigación, los conceptos, alternativas y tendencias que enmarcan las teorías acerca de recuperación de información.

Partiendo de este marco de referencia, finalmente se describe la solución a las necesidades expresadas en un principio, de tal forma que sea puesto a prueba por los distintos usuarios en la Universidad Técnica de Delft, para que sirva como referencia a las futuras investigaciones en el área de recuperación de información que puedan llevarse a cabo al interior de la Universidad EAFIT u otras instituciones de la ciudad.

2 GLOSARIO

Clase: uno de los componentes principales de la Programación Orientada a Objetos (POO), que se encarga de definir los atributos que describen una entidad y las respectivas funciones (métodos) para realizar operaciones sobre dichos atributos.

Content Management Systems (CMS): productos de software que se encargan de almacenar, indexar y presentar distintos tipos de información, utilizado generalmente por grupos de trabajo específicos con la intención de poseer todas las producciones intelectuales recopiladas en un sitio accesible mediante redes informáticas.

Ez Publish: software desarrollado por al empresa EZ SYSTEMS. Este es el gestor de contenidos (CMS) elegido por el personal encargado de administrar la información en la Universidad Técnica de Delft y por tanto el sistema sobre el cual se desarrollará el presente proyecto.

Hilos (Thread): herramienta que proveen algunos lenguajes de programación, que permite la ejecución del mismo segmento de código de manera simultánea.

Internet: también conocido como WWW (World Wide Web). Es una red de servidores que sirven como repositorio de datos a nivel mundial, que permite acceder a la información en cualquier lugar del mundo utilizando unos estándares definidos para su presentación.

Intranet: una red de servidores que sirven como repositorios de datos para una organización específica. La información allí contenida es de propiedad de dicha organización y en la actualidad es un aliado a los procesos empresariales.

Método: parte de un código de software encargado de realizar una función específica dentro de una determinada aplicación.

Página Web: es un archivo de texto que está escrito en un lenguaje que un servidor Web puede interpretar y que es visualizado por un explorador de Internet (Internet Explorer, Netscape, Mozilla, etc.). Este archivo puede ser estático (HTML) o dinámico (PHP, ASP, JSP, JSCRIPT, etc.)

PHP: lenguaje de programación enfocado a desarrollos Web.

Servidor: equipo de cómputo con características específicas, que permite un alto desempeño y seguridad para prestar diferentes tipos de servicios a los usuarios finales. Algunos tipos de servidores son: Web, Correo, Almacenamiento.

Socket: mecanismo de comunicación entre procesos (IPC Inter. Process Comunication). Es una herramienta de programación soportada por los sistemas operativos en la que procesos independientes pueden intercambiar flujos de datos con el fin de realizar funciones específicas con la información recibida del otro proceso.

Synset (Synonym Set): conjunto de palabras sinónimas que utiliza WordNet.

TCP (Transfer Control Protocol): protocolo de transferencia utilizado en redes para transportar flujos de datos entre dos o más puntos ya sea en una máquina o red de computadoras.

Web: término con el que se hace referencia usualmente a Internet.

WordNet: base de datos semántica utilizada para el desarrollo del proyecto.

3 DEFINICIÓN DEL PROBLEMA

La revolución de los computadores y los sistemas han generado un cambio en las concepciones de las personas. Uno de los grandes aportes se ha proporcionado en el sector de la educación, porque han mejorado y/o optimizado los procesos educativos y además han generado una nueva dinámica en el proceso de aprendizaje.

En la facultad de Ingeniería de Diseño de Productos de la Universidad Técnica de Delft, en Holanda, se está llevando a cabo un proyecto que investiga las formas de optimizar el uso de la información en diseño, de tal forma que ayude a que los diseñadores se apoyen en los sistemas de información, y que sus fuentes no se limiten a la interacción tradicional de los libros y colegas. En la actualidad cuentan con una pequeña colección de artículos con información sobre tres temas: bicicletas, elementos de oficina y herramientas de jardinería. Dicha colección se ha creado con el fin de experimentar cuál es el impacto de ofrecer a los diseñadores cierto tipo de información en las etapas tempranas del proceso de diseño.

Con el fin de corroborar el impacto positivo con la publicación de los artículos, se ha decidido realizar una segunda etapa del proyecto y ahora se desea integrar un sistema de búsquedas que brinde soporte al usuario con respecto al tema de investigación, permitiéndole ampliar y reducir los temas de consulta además, de ser posible, aportar sugerencias de acuerdo con las búsquedas realizadas, y las consultas que otros usuarios han realizado y que puedan aportar al proceso. En el presente proyecto se analizará el tema relacionado con la ampliación de los temas a consultar.

Actualmente existe un recurso lingüístico llamado WordNet[1] que brindaría el soporte necesario para que el módulo que realiza las búsquedas en el gestor de contenidos pueda tener una ampliación de sus funciones de una manera eficiente. WordNet es una base de datos léxica inspirada en teorías psicolingüísticas sobre la memoria léxica humana; ésta se basa en árboles de conceptos en los que se pueden definir relaciones semánticas, tales como sinónimos y antónimos, y relaciones léxicas, tales como hiperonimia, hiponimia, meronimia y holonimia. En investigaciones recientes se ha adicionado a cada concepto un dominio, el cual indica a que área del conocimiento pertenece.

La unión de estos dos elementos (sistema de gestión de información y WordNet) será la base para desarrollar la investigación acerca del impacto de las tecnologías de información en las primeras etapas del proceso del diseño.

[1] http://cogsci.princeton.edu/~wn

4 OBJETIVOS

4.1 GENERAL

Desarrollar una solución de software que integre Ez Publish y WordNet de manera que pueda implementarse dentro del sistema de búsquedas de Ez Publish y que permita apoyar los procesos de desarrollo en el diseño de productos que se llevan a cabo en la Universidad Técnica de Delft en Holanda, y que interactúen con el sistema de gestión de información que poseen actualmente.

4.2 ESPECÍFICOS

Analizar y definir una solución basada en un producto de software que permita la integración de las dos herramientas, Ez Publish y WordNet.

Diseñar e implementar un mecanismo que permita al sistema de manejo de contenidos presentar sugerencias de nivel semántico al momento de realizar la consulta de acuerdo a las necesidades y directrices del usuario.

Presentar el sistema de búsqueda en una interfaz de entorno Web que sea amigable para los usuarios.

Realizar la documentación apropiada del proyecto con el fin de que este pueda ser instalado sin ningún percance en la Universidad de Delft.

5 MARCO TEÓRICO

5.1 COMPUTACIÓN SEMÁNTICA

5.1.1 ¿Qué es computación?

Computación o informática se define como los conocimientos y técnicas de cualquier proceso que puede ser realizado por intermedio de un equipo de cómputo de manera autónoma[2]. Todo el procesamiento de las peticiones, realizadas por el usuario final, son completamente realizadas por las unidades de procesamiento de dicho equipo y aplicando reglas ya definidas (algoritmos), se obtienen los resultados al procesar los diversos datos.

En el desarrollo de esta área se ven involucrados conocimientos de electrónica, matemáticas, lógica, manejo de información y comportamiento humano. Todas estas disciplinas se combinan para permitir que las personas puedan tener una mejor adaptación y mayor capacidad de uso de los sistemas, el proceso de ingeniería de software dará unos resultados, los cuales serán desarrollados a partir del diseño principal, de tal forma que sean comprensibles y con una lógica similar a la forma de pensar humana.

5.1.2 ¿Qué es semántica?

La semántica es el campo de la lingüística que se encarga de estudiar el significado de los símbolos y de cómo influye en lo que las personas dicen[3].

---

[2] Significado de Informática, Real Academia Española de la Lengua, http://www.rae.es/

[3] ¿Qué es semántica?, Sergio Zamora http://www.geocities.com/sergiozamorab/semantic.htm

A partir de este concepto es posible derivar un gran número de relaciones que afectan directamente en las maneras de comunicación y expresión que poseemos los humanos.

5.1.3 ¿Qué es computación semántica?

La computación semántica es la integración del procesamiento automático de la información con reglas específicas que toman en cuenta alternativas lingüísticas con el ánimo de generar resultados acordes al raciocinio humano. La programación de alternativas con una mayor adaptación al pensamiento humano es un área estudiada por la Inteligencia Artificial.

5.1.4 Ventajas de la Computación Semántica

- Rapidez
- Confiabilidad
- Diversidad de las fuentes de consulta
- Resultados mas apropiados que los sistemas de coincidencia sintáctica.

La principal ventaja de este modelo de computación es que los resultados obtenidos se acercan más a las necesidades de los usuarios, ya que la adición de las diferentes características del pensamiento, asociación y procesamiento de información realizada por humanos; en el campo computacional pueden proporcionar resultados más diversos, el sistema puede realizar el procesamiento en varias fuentes de información de manera sincrónica, además el proceso es notablemente más rápido.

5.2 XML (EXTENSIVE MARKUP LANGUAGE)[4]

Se define como un “metalenguaje”, el cual proporciona la facilidad de definir lenguajes concretos para representar documentos, de manera independiente de la forma en que se realizará su presentación.

Los documentos XML tienen una visión lógica y una física. La visión lógica define elementos, declaraciones y comentarios identificados mediante marcas explícitas. La visión física contiene datos procesados y/o sin procesar, identificados también por marcas que definen el esquema de almacenamiento y se anidan con la estructura lógica del documento.

La idea general detrás de XML es que pueda ser generado y leído fácilmente por una máquina, procurando crear un estándar que sea independiente de los idiomas, lenguajes de programación, plataformas y que no presenten mayor complejidad, de manera que pueda ser utilizado como método para el intercambio de información estructurada independiente de las aplicaciones y las plataformas que se utilizan para crearla y manipularla.

Ya que es un lenguaje de representación, para poder producir documentos bien formados es necesario realizar una definición del documento tal que, sea posible validar la conformidad del XML con respecto a su esquema. Para crear un documento bien formado se necesitan las siguientes características:

Que tomado como un todo, el documento cumple la regla denominada "document". Esto quiere decir que:

- Respeta todas las restricciones de buena formación dadas en la especificación

[4] Web Semántica; Betancur Toro, Diana Cristina; Universidad EAFIT; 2003

- Cada una de las entidades analizadas que se referencia directa o indirectamente en el documento está bien formada

Para que un documento se considere bien formado se puede realizar de dos formas.
Definir la estructura del documento XML en un DTD (Document Type Definitions), es decir, definir un lenguaje de marcado propio para la aplicación específica.
La segunda forma, de no utilizar DTD el documento XML en su principio debe declararse como "*standalone"*, entendiéndose como "*standalone"* un documento XML que no está asociado con ninguna definición de documento DTD y que por tanto es sólo válido para la aplicación específica.

5.2.1 Estructura de un documento XML[5]

Para la creación de documentos XML se debe tener en cuenta los siguientes elementos:

Caracteres. Un caracter es una unidad atómica de texto tal y como está especificada por ISO/IEC 10646. Los caracteres legales son tabulador, retorno de carro, avance de línea y los caracteres gráficos legales de Unicode e ISO/IEC 10646.

Construcciones sintácticas comunes. Los procesadores XML tratan de forma diferente a un conjunto de caracteres denominados "espacios en blanco", los cuales son: "espacio" (Unicode/ASCII 32), tabulador (Unicode/ASCII 9), retorno de carro (Unicode/ASCII 13) y salto de línea (Unicode/ASCII 10).

[5] Tomado de Web Semántica; Betancur Toro, Diana Cristina; Universidad EAFIT; 2003

La especificación XML 1.0 permite el uso de esos "espacios en blanco" para hacer más legible el código, y en general son ignorados por los procesadores XML. Igualmente al utilizar XML, es necesario asignar nombres a las estructuras, tipos de elementos, entidades, elementos particulares, etc.

Para asignarles los nombres a estos elementos se debe tener en cuenta que no se puede usar al comenzar el nombre la cadena "xml", "xML", "XML" o cualquier otra variante. Se pueden usar las letras y guiones en cualquier parte del nombre al igual que dígitos, guiones y caracteres de punto, pero no se puede empezar por ninguno de ellos. Caracteres como algunos símbolos y espacios en blanco, no se pueden usar.

Datos de caracter y de marcado. Todo texto está constituido por datos de caracter y de marcación. La marcación toma la forma de tags, etiquetas en español, para el comienzo y finalización, elementos vacíos, referencias de entidad, referencias de caracter, comentarios, delimitadores de secciones CDATA, declaraciones de tipo de documento, e instrucciones de procesamiento. Todo texto que no sea marcación constituye los datos de caracter del documento. En el contenido de elementos, los datos de caracter son cualquier cadena de caracteres que no contenga el delimitador de comienzo de ninguna marcación.

Comentarios: pueden aparecer en cualquier lugar de un documento; Adicionalmente pueden estar en lugares permitidos por la gramática. No son parte de los datos de caracter de un documento.

Instrucciones de procesamiento (IPs): permiten a los documentos incluir instrucciones para las aplicaciones. Las IPs no son parte de los datos de caracter del documento, pero deben ser pasadas a la aplicación.

Secciones CDATA: pueden ocurrir en cualquier lugar en que pueda encontrase un dato de caracter; son usadas para omitir bloques de texto que contengan caracteres que de otro modo serían reconocidos como marcas.

Prólogo y declaraciones de tipo de documento: Aunque no es obligatorio usarlo, el prólogo es la primera línea del documento, donde se indica: la versión de XML a usar en el mismo, la codificación del documento; que depende del parser (descrito anteriormente como procesador de XML, es el encargado de determinar los datos y las etiquetas en el documento) el entender o no la codificación; además de la declaración acerca de la asociación del documento, es decir autónomo (*standalone*) o si usa DTD.

5.3 WEB SEMÁNTICA

5.3.1 ¿Qué es?

Es una gran combinación de información enlazada y codificada en tal forma que puede ser fácilmente procesada por máquinas que involucra reglas de procesamiento semántico[6]. La manera de realizar el procesamiento se lleva a cabo con las distintas etiquetas que se encuentran dentro del código XML, las cuales poseen un contexto determinado para poder procesarlas. La Web semántica pretende añadir a la Web información semántica ya que permitiría la definición de ontologías[7], estas permitirán realizar asociaciones entre conceptos (ver nota al pie).

En la actualidad existen grandes esfuerzos para lograr definir un estándar[8] que pueda ser utilizado a nivel mundial para cualquier tipo de comunicación dinámica entre aplicaciones en donde incluso el lenguaje del contenido no sea obstáculo. Dicho estándar daría la libertad de tener las estructuras, los datos en distintos lenguajes y estos podrían ser utilizados y/o referenciados desde cualquier otro sin esta restricción. Para un ejemplo más actual es posible referenciar a UML como estándar para el desarrollo de software.

Para poder completar el modelo de Web semántica, es necesario además de utilizar XML, hacer uso de algunos módulos adicionales que permitirán definir las

---

[6] The semantic web: an introduction, http://infomesh.net/2001/swintro/

[7] En ciencias de la computación el término ontología se utiliza para referirse a estructuras de representación del conocimiento que permiten realizar algunas inferencias automáticamente. No debe ser confundido con la parte de la metafísica que trata del ser en general y de sus propiedades trascendentales.

[8] World Wide Web Consortium, http://www.w3.org/

abstracciones de una forma adecuada y relacionar dichas abstracciones. En síntesis, es necesario definir un modelo para describir de la manera más estricta posible un campo de conocimiento determinado, para que al ser utilizado por una aplicación, se disminuyan los riesgos de dualidad de la información (que no se combinen los conceptos entre campos de conocimiento diferentes), obteniendo, por tanto, mejores resultados.

Técnicamente este proyecto no es una aplicación que pueda incluirse dentro de la Web Semántica, ya que el núcleo que compone la lógica de programación de la aplicación aun no está desarrollado para cumplir con el estándar XML (específicamente con las respectivas declaraciones de datos (DTD)) de manera que pueda intercambiar información con otras aplicaciones. Se menciona el tema de Web Semántica para referenciar los desarrollos que en este momento se están realizando con WordNet, con el fin de acoplarlo a la tendencia que trae XML y poder desarrollar sistemas encaminados al intercambio de información con este estándar.

5.3.2 Ventajas de la Web Semántica

Integración de información independiente de la plataforma o del idioma, gracias a la definición de las ontologías, ya que con el contexto definido, las reglas para comprender el significado de una palabra determinada dentro de dicho contexto serán iguales en cualquier lugar.

5.4 INTELIGENCIA ARTIFICIAL

Es el campo de la computación encargado de modelar, crear e implementar modelos de software que se asemejen a los distintos comportamientos humanos. Entre las múltiples ramas de la Inteligencia Artificial (IA) podemos destacar las siguientes (que son las que más relevancia tienen con el presente proyecto):

5.4.1 Procesamiento de Lenguaje Natural (PLN)

Son elementos de software que tienen la capacidad de sobrellevar una conversación con un ser humano de manera interactiva. Implican el uso de técnicas de la computación semántica.

5.4.2 Redes Neuronales

Son elementos de software muy simples, programados de tal forma que asemejan las conexiones cerebrales. Dichas relaciones son basadas en conexiones entre neuronas a las cuales se les asigna reglas para definir con cual de las neuronas adyacentes debe relacionarse con el fin de obtener la solución al problema.

5.5 SISTEMAS MANEJADORES DE CONTENIDOS (CMS – CONTENT MANAGEMENT SYSTEMS)

5.5.1 ¿Qué es contenido?[9]

Debido a que las instituciones tienen un gran volumen de información, se ha hecho necesario administrarlo de una manera óptima y de fácil acceso para las personas interesadas. El método más utilizado para recopilar las producciones es a través de documentos; la problemática que trae el manejo de documentos es que lo toma como un todo "absoluto" y no precisa que en su interior pueden existir diversos temas que no son expresamente de relación con el título del documento.

Contenido es en esencia, cualquier tipo de información (digital en este caso) que puede ser utilizado para poblar una página Web. Pueden incluirse textos, imágenes, videos, en general todo tipo de información que puede ser publicada a través de Inter/Intra o Extranet.

Es apropiado entonces crear un modelo de asociación que permita seleccionar toda la información de los documentos de una forma más simple.

5.5.2 ¿Qué son los CMS o sistemas gestores de contenidos?

Un sistema manejador de contenidos es un software que permite asociar la información que se encuentra en artículos y que por el nombre del documento no es posible relacionarla con un tema o campo de conocimiento específico.

---

[9] http://www.contentmanager.eu.com/document.htm

Un CMS es una herramienta que provee facilidades de diseño, integración y adición de contenidos de manera automática a un sitio en la red (Internet, Extranet o Intranet).[10] Su filosofía se basa en utilizar plantillas predeterminadas para los diseños gráficos, utilizar bases de datos para almacenar los contenidos y presentar una interfaz vía Web que permita interactuar con el CMS (tanto para administración como para la adición del contenido) sin la necesidad de estar presentes en el servidor o tener conocimientos de lenguajes de programación para visualización en browsers (html, php, xml, etc.).

5.5.3 ¿Cuales son sus ventajas?

- Separa el contenido, la estructura y el diseño de los sitios Web.
- Facilidad de producción de contenido sin requerir habilidades de programación.
- Mantenimiento descentralizado, múltiples personas pueden administrar el sistema remotamente.
- Conserva la consistencia del diseño, no es necesario preocuparse por el formato de presentación de los contenidos, el sistema se encarga de eso.
- Generación automática de la navegación basándose en el contenido de la base de datos, evitando referencias a páginas no existentes.
- Permite cooperación entre los autores.
- Control de la producción y autoría de los contenidos publicados.
- Control de usuarios para la publicación o vista de los distintos contenidos.
- Fácil adaptación a cambios de imagen del sitio mediante las plantillas. Un cambio en un archivo es aplicado al todo el sistema.
- Brinda facilidades de flujo de trabajo con el fin de adicionar nuevas funcionalidades al CMS sin necesidad de reprogramar todo el sistema. Es

[10] http://www.contentmanager.eu.com/history.htm

posible tener la figura de un “director de publicación” quien podrá autorizar que los contenidos sean o no mostrados a los usuarios.

5.5.4 EZ PUBLISH

Es un CMS desarrollado totalmente en lenguaje PHP y con un enfoque orientado a objetos que permite realizar diversas adiciones a su núcleo básico con el fin de adaptar el sistema a las necesidades específicas.

Ez Systems, la compañía productora del CMS, tiene como principio del desarrollo, programar el sistema en forma de módulos (es una ventaja ya que cambios y/o adiciones pueden realizarse sobre el sistema sin complicaciones). Cada módulo cuenta con distintas clases que realizan las distintas funciones. Con este modelo pretenden que las mejoras y adiciones al sistema sean más transparentes y que a su vez sean independientes logrando así un menor desgaste al momento de incorporar dichos cambios o adiciones.

La elección de este manejador fue realizada en Delft. En realidad los motivos por los cuales eligieron este CMS en específico son desconocidos. Para el presente proyecto sólo se informó que este era el sistema a utilizar. Ez Publish posee un módulo que presenta interfaz con XML, que es una de las alternativas que más se está utilizando en la actualidad para intercambio de información independiente de la plataforma y con diferenciación de la información gracias a las etiquetas que utiliza el lenguaje.

5.6 BASES DE DATOS SEMÁNTICAS

5.6.1 WORDNET[11]

En la actualidad se ha generado la necesidad de tener fuentes de información que puedan utilizar el beneficio que brinda la computación. Para el tema de este proyecto en específico, la necesidad de un "diccionario" en formato digital con su correspondiente interfaz es fundamental.

Para los creadores de WordNet el simple hecho de diseñar unos algoritmos que realicen una búsqueda secuencial en unos archivos no era importante; lo que ellos pretenden es que la herramienta sea una mezcla efectiva de la combinación léxico-gráfica tradicional y la computación de alta velocidad moderna.

WordNet se define como una referencia léxica inspirada en teorías psicolingüísticas de la memoria léxica humana (es un análisis de la forma en que los humanos asocian los términos). Organiza verbos, adverbios, sustantivos como conjuntos de sinónimos (Synset) donde cada uno representa un concepto léxico. Cada Synset se relaciona con otros por medio de relaciones semánticas.

A continuación se hará una breve referencia de las relaciones utilizadas en WordNet.

---

[11] The 5 papers. George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Millar
[18] WordNet: An electronic lexical database

- Sinonimia

Este tipo de relación es la que indica que un par de palabras tienen el mismo significado dentro de un mismo contexto, de manera que, al reemplazar una por la otra el sentido de la oración permanezca intacto. Debido a que la unidad de trabajo básica de WordNet es el concepto (representado mediante un synset), la relación de sinonimia no está explícitamente representada en WordNet; todas las palabras que pertenecen a un mismo synset se consideran sinónimas. Es importante notar que los tipos de divisiones primarias en WordNet conllevan a subdividir los synsets de acuerdo al tipo de palabra que pertenecen, es decir, dividirlos en sustantivos, verbos, adverbios y adjetivos.

- Antonimia

La forma más sencilla de definir un antónimo es cuando podemos decir que una palabra es contraria a otra; lo cual es correcto. En la actualidad existe una dificultad ya que las relaciones de antonimia no son excluyentes; para explicarlo puede ser útil el siguiente ejemplo: Las palabras *rico* y *pobre* son antónimas, pero el hecho de decir que una persona no es pobre, necesariamente no quiere decir que ser calificada como rica y viceversa.

Algunas personas consideran la antonimia como una relación léxica entre las formas de las palabras y no una relación semántica entre los significados de las palabras. Por esta razón, en WordNet al especificar la relación de antonimia se detalla cuales son las palabras involucradas en la relación.

- Hipónimos e Hiperónimos

Es una relación semántica entre palabras que denota subordinación. Según la Real Academia Española[12], un hipónimo es una palabra cuyo significado está incluido en otra. Para comprenderlo de una forma más sencilla, si tenemos dos

---

[12] http://www.rae.es

palabras *x* y *y*, *x* es el hipónimo de *y* si se sobreentiende que *x* es un tipo de *y*. La relación podría expresarse de la siguiente manera, una *motocicleta* es un tipo de *vehículo*. Y de la forma contraria *y* es el hiperónimo de *x*.

- Merónimos y Holónimos

Es una relación semántica que denota pertenencia;[13] las construcciones son similares a: *x* es una parte de *y*, en este caso *x* es un merónimo de *y*, e *y* es el holónimo de *x*. En el caso de WordNet se distinguen tres modalidades de merónimos (con sus modalidades de holónimos equivalentes). Merónimos de sustancia que definen de que material está constituido algo (ej. El cuerpo humano está constituido por células). Merónimos de pertenencia, que definen una relación de pertenencia a un grupo, ser miembro de una comunidad (ej. Cada individuo es miembro de una sociedad). Merónimos de composición, que define ser un componente de algo (ej. Un computador está compuesto de tarjeta madre, procesador, disco duro, etc.).

[13] Aunque la comunidad académica española utiliza los términos merónimo y homónimo para referirse a las relaciones semánticas que denotan composición, la Real Academia Española no incluye estas palabras en su diccionario.

5.7 RECUPERACIÓN DE INFORMACIÓN (INFORMATION RETRIEVAL)[14]

Es la disciplina que se encarga del análisis de los modelos que representan, almacenan y organizan datos, de tal manera que proporcionen un acceso sencillo a la información solicitada. Se caracteriza por trabajar con datos no estructurados, a diferencia de las técnicas de bases de datos.

El principal objetivo de esta área es facilitar a los usuarios encontrar información relevante con respecto a un tema propuesto de acuerdo a la formación léxica y semántica de la búsqueda como tal; es decir: no es simplemente realizar una búsqueda y presentar los resultados que concuerdan específicamente con las palabras de búsqueda, sino que además ampliará la búsqueda a temas relacionados con las palabras principales.

5.7.1 Conceptos básicos

Tareas de Usuarios

"Se considerará como tarea de usuario, toda actividad que cualquier persona realice con el fin de obtener información más detallada con respecto a un tema."[15]
En este punto es necesario aclarar que hay dos tipos de tareas de usuarios.
1. Observación de información (*Browsing*): es aquel proceso que la persona inicia sin una definición exacta de los elementos que desea buscar, por tanto, es posible que pueda desviarse de la consulta primera al encontrar otros temas que capten su interés.

---

[14] [16] WITTEN, Ian H. Managing Gigabytes

[15]]16] WITTEN, Ian H. Managing Gigabytes

2. Recuperación de información: es cuando la persona tiene una definición clara del tema a consultar y se ceñirá a los resultados que contengan dicha búsqueda.

Visión lógica de los documentos

Históricamente la representación de los documentos se ha realizado con base en un conjunto de palabras claves (*keywords*). La visión lógica de los documentos es un conjunto de términos (palabras claves o keywords) que describen concisamente el contenido de cada uno. Para obtener dichas claves es necesario aplicar uno de los siguientes procedimientos. Primero extraerlas directamente desde el texto del documento mediante un algoritmo, o segundo que sean escritas por un ser humano que sintetice lo dicho allí; independiente del método utilizado, este definirá una visión lógica del documento.

Teniendo en cuenta la funcionalidad que tendrá el sistema y el diseño de almacenamiento y procesamiento de los datos, es posible hacer uso de las capacidades de procesamiento de los sistemas de cómputo actuales para generar las visiones lógicas de los documentos que va a contener.

Es posible tener la representación lógica total del documento; este sistema brinda los mejores resultados al realizar búsquedas pero es el más ineficiente por el espacio de almacenamiento que necesita para albergar toda la colección de documentos y sus índices. También es posible realizar algunos procedimientos para obtener un conjunto reducido de términos que representen el contenido.

Para lograr reducir estas representaciones en largas colecciones de documentos y tener un conjunto de palabras claves menos extenso, pueden utilizarse reglas (preprocesamiento) para eliminar palabras de parada (stopwords como: artículos y conectores; ej. de, el, para, con, etc.), el uso de stemming (colocar las palabras en

su raíz gramatical o forma original; ej. conjugación → conjugar), o grupos de sustantivos (reduce los verbos, adverbios y adjetivos).

Estas transformaciones reducen la complejidad de la representación del documento, reduciendo la visión lógica del documento completo a un conjunto de términos claves pero perdiendo información en el proceso. Además de estas técnicas, también es posible aplicar algoritmos de compresión a los documentos, para continuar en la tarea de reducir el espacio de almacenamiento requerido para dichas colecciones; el inconveniente que pueden presentar estas técnicas es que los tiempos y la capacidad de cómputo requerida para realizar dichos procedimientos sea bastante alta ó que los algoritmos utilizados para recuperar la información excedan los tiempos de respuesta requeridos por el sistema. Generalmente los tiempos de respuesta pueden ser evaluados por los usuarios finales quienes compararán la calidad de los resultados obtenidos en el tiempo que toma el proceso. La tendencia en este sentido es utilizar esquemas de compresión que permitan realizar la búsqueda sobre el texto comprimido[16]. Finalmente, es interesante mencionar que si algunos documentos de la colección son confidenciales y hay que controlar el acceso a ellos, se puede acudir a la criptografía para cifrarlos. Sin embargo, también hay que garantizar que no se pueda inferir su contenido o existencia a partir de los índices y de las respuestas a las consultas. Este tema es otra línea de investigación muy importante.

5.7.2 Consideraciones de los modelos de recuperación de información

Antes de enunciar los modelos, es importante conocer las dos tendencias que existen en la actualidad con respecto al comportamiento de las búsquedas y los documentos.

[16] Managing Gigabytes; Witten, Ian; Moffat, Alistair; Bell, Timothy; Morgan Kaufman Publishers.

Ad hoc: se presenta cuando las colecciones de documentos permanecen relativamente estáticas y las búsquedas que ingresan al sistema son muy variables. La mayoría de las tareas de búsqueda en la actualidad siguen este patrón.

Filtering: se presenta cuando las opciones de búsqueda son relativamente estáticas, pero los documentos son muy variables. Es bastante común observarlo en el mercado accionario.

A manera de caracterización, los modelos de recuperación de información deben cumplir con lo siguiente:
Un modelo de recuperación de información es una cuádrupla [**D**,**Q**,F,R(q_i,d_j)] donde[17]:
D es el conjunto de vistas lógicas (representaciones) de los documentos en la colección
Q es el conjunto de vistas lógicas de las necesidades de los usuarios. Dichas representaciones son llamadas "queries" (consultas).
F es un Framework (entorno) que modela las representaciones de los documentos, las búsquedas (queries), y sus relaciones.
R(q_i,d_j) es una función de clasificación que asocia un número real con una búsqueda q_i Є Q y la representación del documento d_j Є D (que tan relevante es el documento de acuerdo al número asignado). Este ranking define un orden entre los documentos en relación con la búsqueda q_i.

Para el alcance del presente proyecto, enunciaremos las generalidades de los modelos clásicos de recuperación de información.

[17] Modern Information Retrieval, capítulo 2 Modeling.

5.7.3 Modelos clásicos de recuperación de información

La premisa que utilizan los modelos clásicos es que cada documento esta descrito en un conjunto de palabras representativas, llamados *index terms*. Un *index term* es simplemente una palabra cuya semántica ayuda a recordar los temas principales del documento. Puesto que estos términos son utilizados como sumario, las palabras más descriptivas para este grupo son los sustantivos, ya que tienen significados por si solos.

Ya obtenidos los términos claves (index terms), se procede a asignar un valor numérico (llamado el peso) que determina la relevancia de la palabra dentro del documento. Cabe anotar que encontrar las palabras claves no es una tarea fácil, ya que debe ser lo suficientemente descriptiva con respecto al contenido y al mismo tiempo no ser muy común de tal forma que un gran número de documentos la tengan, dado que este hecho restaría importancia a la palabra al ampliar la cantidad de documentos de la colección que serían recuperados con este término.

Modelo Booleano

Este es un modelo simple basado en el álgebra booleana. Las búsquedas son especificadas como expresiones booleanas con una semántica precisa. El modelo considera que los términos están o no presentes en un documento, como resultado los pesos son valores booleanos. La búsqueda es representada en una forma normal disyuntiva (FND). El algoritmo de búsqueda encuentra la relación entre los parámetros de entrada y los documentos, si el documento es relevante la función retornará 1, de lo contrario retornará 0.
Cuando el modelo retorna que un documento es relevante, se puede tener total seguridad de que los parámetros de búsqueda fueron encontrados exactamente,

este modelo no admite resultados parciales; lo cual es una de las grandes desventajas que tiene, además puede considerarse más como un modelo de recuperación de datos que como modelo de recuperación de información.

Modelo vectorial

Considerando la gran desventaja del modelo Booleano, el modelo vectorial asigna pesos no booleanos a los términos, de tal forma que concordancias parciales son posibles de obtener como resultados. Otra de las ventajas de este modelo es que son usados para calcular los grados de similitud de los documentos. El algoritmo de búsqueda utiliza varios vectores, el primero es el vector de la búsqueda que contiene los pesos de la búsqueda con respecto a los términos claves del sistema, los siguientes vectores contienen los términos claves de cada documento y sus pesos. Se hace un análisis de correlación entre los vectores de los documentos y el vector de búsqueda, aquellos con mayor correlación serán los elegidos para presentar al usuario.

Modelo probabilístico

Este modelo procura encontrar las probabilidades que un determinado documento satisfaga las necesidades de búsqueda del usuario. Asume que existe un conjunto de documentos que son relevantes con respecto a la búsqueda, por tanto los demás documentos se considerarán no relevantes. Ya que en un principio no se tiene referencia del conjunto de documentos relevantes, se asignan unas probabilidades estáticas con el fin de obtener un subconjunto parcial que contenga términos claves. A partir de allí se define un límite que será la base para ir depurando el conjunto de resultados; después de un proceso iterativo en el que

las probabilidades cambian entre cada subconjunto se halla la función de similitud entre la búsqueda y los términos claves obtenidos.

La selección de las palabras claves se puede hacer automática o manual. La técnica más utilizada para determinar las palabras claves de manera automática es contar la frecuencia de aparición de un término y asociar su importancia con su frecuencia. La manual corresponde a la acción que realiza una persona con suficiente conocimiento del tema para según su criterio, conformar el conjunto de palabras claves para cada documento.

Mediciones de los resultados

Por otra parte, el resultado de la búsqueda se puede medir en términos de precisión y cubrimiento. La precisión es el porcentaje de documentos recuperados que tienen que ver con la consulta. El cubrimiento es el porcentaje de documentos recuperados que son relevantes con la consulta.

5.7.4 LENGUAJES DE BÚSQUEDA

Teniendo mayor conciencia de que el proceso de recuperación de información no consiste simplemente en efectuar una búsqueda y que una máquina nos retorne unos resultados, desde el punto de vista del diseño y la programación vale tomar en consideración la necesidad de definir el mecanismo con el cual los programas y los usuarios pueden interactuar.

Ya que existen diferentes modelos de recuperación de información, no es posible considerar que una misma forma de formular las búsquedas sea igual de exitosa en cada uno de los modelos, por esto, además de saber con que algoritmos se

realizarán los procesos, es necesario establecer una interfaz de tal forma que usuarios y maquinas puedan tener un lenguaje común de comunicación en el que eficientemente se especifiquen las necesidades por parte de los seres humanos, que pueda acoplarse al sistema para ser resuelto con el modelo elegido.

A continuación se enunciarán algunas características de los lenguajes de búsqueda más utilizados.

5.7.4.1 Búsquedas de una sola palabra

Es la forma más común y sencilla de realizar una consulta. Simplemente se ingresan una o varias palabras a la búsqueda y el sistema busca en las bases de datos cuales documentos contienen cada una de las palabras por separado. Los resultados se retornan de acuerdo a la cantidad de términos que están en la búsqueda y en el documento. Es posible que algunos sistemas acepten concordancias parciales, de forma que complementen los resultados obtenidos.

5.7.4.2 Búsquedas de contexto

Ya que la búsqueda de palabras no toma en cuenta si los términos aparecen de forma consecutiva sino que determina el número de apariciones en el texto, las búsquedas de contexto procuran reducir el espacio de resultados de acuerdo a los términos alrededor de las palabras que desean consultarse. Las consultas de frases toman en cuenta que la frase es una secuencia de palabras y lo que procuran es encontrar dichos términos de manera consecutiva, quitando los caracteres separadores y/o palabras sin importancia (usualmente llamadas “conectores”). El método de proximidad da una mayor holgura al definirse una regla de caracteres o palabras en las que se puede encontrar la ocurrencia de la

siguiente palabra de búsqueda. Es decir, dada la ocurrencia del primer término, tomando la configuración del sistema, buscará si en un espacio determinado (20 caracteres, 4 palabras) se encuentra la siguiente palabra a buscar. Al referirse a frases opera de la misma forma con cada uno de los términos que componen la frase.

5.7.4.3 Búsquedas booleanas

Basada en la simple lógica booleana, este modelo se compone de una estructura sencilla, que además, puede reaplicarse sobre los resultados obtenidos anteriormente. El tipo de la búsqueda es de este estilo e_1 *EXP* e_2 donde los valores e_1 y e_2 son los términos a comparar y la expresión (*EXP*) corresponde a uno de los siguientes operadores:

- OR: la búsqueda e_1 OR e_2, trae los elementos que cumplan con cualquiera de las expresiones.

- AND la búsqueda e_1 AND e_2, trae los elementos que cumplan con ambas expresiones.

- BUT la búsqueda e_1 BUT e_2, trae los elementos que cumplan con la expresión 1 y que no cumplan con la expresión 2.

5.7.4.4 Lenguaje natural

En este modelo se aprecia como el algoritmo asigna pesos (valores numéricos) a los resultados obtenidos en la búsqueda. El algoritmo evalúa el grado de similitud del texto en la búsqueda con respecto a los distintos índices de los contenidos,

donde aquellos con mayor semejanza tendrán los pesos con valores más altos, por tanto serán más relevantes los resultados con mayor peso dentro de todo el conjunto de resultados. Para obtener una reducción de este conjunto, es necesario establecer una “política de barrera”, la cual realizará un análisis de los pesos obtenidos y aquellos que se encuentren por debajo de cierto nivel, se considerarán no relevantes y por tanto serán omitidos del resultado final presentado al usuario.

5.7.5 OPERACIONES SOBRE LOS TEXTOS

A parte de la discusión acerca de reducir las longitudes de los textos y las palabras significativas para representarlos dentro de una colección, tomando en cuenta la posibilidad de que dicha reducción pueda llevar a confundir a los usuarios acerca de la forma como se realiza la búsqueda, se ha encontrado que existen algunas reglas que permiten formar de manera automática un conjunto de términos claves más reducido y posiblemente con mayor eficiencia para algunos sistemas.

El preprocesamiento del texto es la alternativa para lograr esto, y es un procedimiento que es posible dividir en cinco subtareas.

5.7.5.1 Análisis léxico del texto

En esta primera fase, el sistema se encarga de convertir el documento de un flujo de caracteres a un flujo de palabras que pueden ser representativas. Para lograr dicho objetivo es necesario realizar algunos cambios al texto original como: suprimir números, espacios extras y signos de puntuación. La discusión que se genera desde este punto es la de precisar si es conveniente aplicar dichos cambios, por ejemplo si es un sistema en los que las fechas son factores determinantes para la búsqueda (historial de acciones en la bolsa, colección de

documentos de eventos históricos, etc.) no es apropiado suprimir esta información. Similar es el trato con palabras que contienen signos de puntuación en su interior y que no delimitan el final o comienzo de una nueva frase.

5.7.5.2 Eliminación de términos de parada

Existen algunos términos que son bastante frecuentes en la escritura normal y que por esta razón se consideran como no relevantes para diferenciar un documento dentro de una colección. Se han hecho estudios que eliminando los términos de parada, es posible reducir la estructura de indización de un documento hasta un 40%. Los conjuntos de palabras que usualmente han sido considerados como términos de parada son: artículos, preposiciones, conjunciones, algunos verbos, adverbios o adjetivos también pueden considerarse allí.

Dependiendo del tipo de sistema que vaya a ser implantado, cada una de estas técnicas de preprocesamiento y procesamiento genera algún tipo de discusión. Si la colección de documentos presenta la posibilidad que las búsquedas sean de frases específicas, por ejemplo en un sistema de obras literarias, en donde la mayoría de las búsquedas serán frases célebres de los escritores.
La frase exacta puede contener en su mayoría artículos, adverbios, etc. que son utilizados como juegos de palabras del escritor, pero para el sistema simplemente pueden ser términos no apropiados para realizar la descripción del texto basado en las técnicas anteriormente descritas.

5.7.5.3 Stemming

A parte de reducir el conjunto de palabras claves para identificación, esta técnica transforma las palabras en su raíz, quitando prefijos y sufijos. Si esta técnica es

utilizada en el sistema, será necesario realizar el stemming con las palabras ingresadas al momento de realizar la búsqueda para que de esta forma puedan retornarse resultados relevantes.

5.7.5.4 Selección de términos para el índice

Considerando que se haya decidido en la planeación del sistema que los documentos tendrían un preproceso para generar su respectivo índice, el paso final es seleccionar los términos que efectivamente representan la idea del documento y que lo hacen diferente a los demás documentos en la colección.

Para documentos científicos, probablemente algún experto en la materia que haya leído el documento podrá realizar el índice que lo identificará; a pesar de esto, la idea para éstos y otros tipos de documentos es que los equipos de cómputo puedan realizar este paso basados en algunas reglas. Otra posible estrategia es tomar todos los sustantivos en su forma simple, eliminando las repeticiones de estos, dejando un conjunto de sustantivos únicos en cada documento. Algunos términos pueden ser compuestos (usualmente dos sustantivos) y juntos pueden tener más significado que separados, considerarlos como un solo término puede ser muy apropiado. Estos son algunos ejemplos que enuncian algunos tipos de estrategias a utilizar en la creación de los índices.

5.7.5.5 Tesauros

Esta es una técnica que consiste en formar un conjunto de palabras guía que serán consultadas por el proceso que genera los índices. El proceso es sencillo: si la palabra existe en el tesauro será utilizada para el índice, de lo contrario será descartada. El tesauro puede ser una muy buena herramienta, sin embargo el

cuidado que debe tenerse es que los temas tratados en la colección pertenezcan al mismo campo del conocimiento para que, partiendo del mismo contexto, los términos claves sean apropiados para la descripción de los documentos.

5.7.6 ALGORITMOS DE BÚSQUEDA SECUENCIAL

Considerando el tipo de proyecto y el tipo de búsqueda que maneja el gestor de contenidos, se reducirá el tema de algoritmos de búsqueda a los más utilizados en ambiente Web ya que las posibilidades de almacenamiento y procesamiento actuales, permiten que no se construyan estructuras de datos como índices para realizar las búsquedas.

5.7.6.1 Fuerza Bruta

Dada una cadena como patrón de búsqueda, el algoritmo trata de buscar todas las ocurrencias del patrón dado en el los textos, este algoritmo no requiere preprocesar el patrón.

Algunas modificaciones del algoritmo usan un esquema un poco distinto. Toman la longitud del patrón como una ventana que se desliza sobre el texto, el chequeo se realiza para ver si las palabras dentro de la ventana coinciden con el patrón, de coincidir, la posición de la ventana es reportada como acierto en la búsqueda, luego la ventana continua deslizándose.

5.7.6.2 Knuth-Morris-Pratt[18]

También conocido como KMP, este algoritmo utiliza ventanas deslizantes sobre el texto, con la diferencia que no ensaya todas las posiciones. El algoritmo analiza que parte del patrón coincide con la ventana y a partir de allí puede saltar algunas posiciones que no serán necesarias de chequear ya que coinciden con el patrón.

Analiza la ventana construyendo prefijos y sufijos cuando la comparación no es perfecta; los prefijos indican hasta que posición de la ventana el patrón coincide, tiene demarcada cual es la posición de la letra que no se ajusta, a partir de la posición anterior comienza a construirse un sufijo que también será comparado con el patrón de forma que, de coincidir, la ventana podría desplazarse hasta dicha posición y ahorrar comparaciones en caracteres anteriores.

Al interior de la ventana se maneja un apuntador que avanza a la siguiente posición cada vez que la letra concuerda con la posición, de alcanzar el final de la ventana, se reporta el éxito de la búsqueda. De no coincidir, la ventana avanza hasta la posición marcada por el sufijo, pero el puntero no cambia de posición en el texto.

5.7.6.3 Familia Boyer-Moore[19]

Este algoritmo, como el anterior, utiliza una ventana que se va deslizando sobre el texto. Difieren en que éste realiza la comparación del patrón desde el final de la ventana hacia el principio de la misma. Al tomar la ventana, el algoritmo compara la última letra de la ventana con la del patrón, de coincidir el algoritmo continuará

---

[18] La definición de este algoritmo puede encontrarse en el URL: http://www.nist.gov/dadt

[19] La definición de este algoritmo puede encontrarse en el URL: http://www.nist.gov/dadt

comparando la anterior letra del patrón y de la ventana hasta encontrar un éxito o una falla en el proceso; de no coincidir el algoritmo compara la última letra de la ventana con la penúltima letra del patrón, con el fin de buscar si dentro de la ventana existe una porción que coincida con el patrón buscado.

Cuando ninguno de los dos procedimientos funciona, el algoritmo salta en el texto de dos formas. Primera, si no existe alguna ocurrencia del patrón dentro de la ventana, la ventana saltará hasta la ultima posición mas uno de la ventana actual. Segunda, de existir la ocurrencia, la ventana posicionará el prefijo encontrado alineado con el patrón buscado y se procederá a realizar la comparación desde el último caracter de la ventana, como se describió al principio.

5.8 ANÁLISIS DE ELEMENTOS DEL MARCO TEÓRICO APLICADOS AL PROYECTO

Ez Publish puede describirse como un Sistema Gestor de Contenidos en el que la creación de índices utiliza todas las palabras contenidas en cada uno de los documentos que se ingresan al sistema, es decir, realiza el análisis léxico para identificar cada palabra y posteriormente las ingresa al tesauro, omitiendo los pasos de eliminación de términos de parada, stemming y la selección de términos para el índice. Al momento de recuperar la información, utiliza un algoritmo para búsquedas de palabra en donde evalúa si cada uno de los términos de búsqueda se encuentra incluido en el tesauro. Aquellos artículos que contengan los términos de búsqueda serán el resultado de la consulta.

La inteligencia Artificial es utilizada con el etiquetador de palabras Tree Tagger utilizado antes de realizar las consultas en WordNet, gracias al Tree Tagger sólo las palabras que no se consideran como *“stop words”* (términos de parada) serán consultadas en la base de datos semántica. Antes que el software pueda funcionar

óptimamente es necesario realizar un entrenamiento con un corpus (colección grande de documentos que está completamente etiquetada en donde cada término tiene su correspondiente tipo de palabra al que pertenece ya sea artículos, verbos, sustantivos, adjetivos, etc). Tree Tagger se basa sobre una red neuronal y esta red es la que se entrena con el corpus.

Con respecto al tipo de aplicación al que puede pertenecer, este proyecto puede considerarse incluido en el campo de la computación semántica puesto que, para formar parte de la Web semántica sería necesario definir un entorno DTD en XML para el marco de la aplicación. El proyecto utiliza algunas variables de entrada en las que se realiza un procesamiento de búsquedas de los términos en los distintos campos semánticos, en donde todos los contextos a los que pertenece la palabra son presentados al usuario final como respuesta de su solicitud.

6 ANÁLISIS DE LA SOLUCIÓN

6.1 REQUISITOS DE LA SOLUCIÓN

Como el presente proyecto se está desarrollando de forma remota, las necesidades que fueron expresadas por el encargado en Delft fueron realizadas a través de correo electrónico; en el **Anexo A** se encuentra una copia de los mensajes.

6.2 SITUACIÓN EN DELFT

El núcleo del sistema es el manejador de contenidos (CMS, Content Mangement System) EZ Publish desarrollado por la empresa EZ Systems que es distribuido con licencia de código abierto GPL[20] (open source), para su funcionamiento es necesario tener los siguientes componentes instalados:

- Servidor Apache versión 1.3 o posterior, es el que permite ver páginas Web (HTML, PHP, XML, etc.)

- PHP, es un lenguaje de programación que tiene compatibilidad completa con el servidor Apache; Ez Publish está completamente desarrollado en este lenguaje.

- Motor de base de datos, es posible tener las siguientes bases de datos: MySql (esta es la instalada) y PostgreSQL.

[20] Proyecto GNU http://www.gnu.org/copyleft/gpl.html

6.2.1 Estructura del gestor de contenidos

El CMS contiene los siguientes módulos:

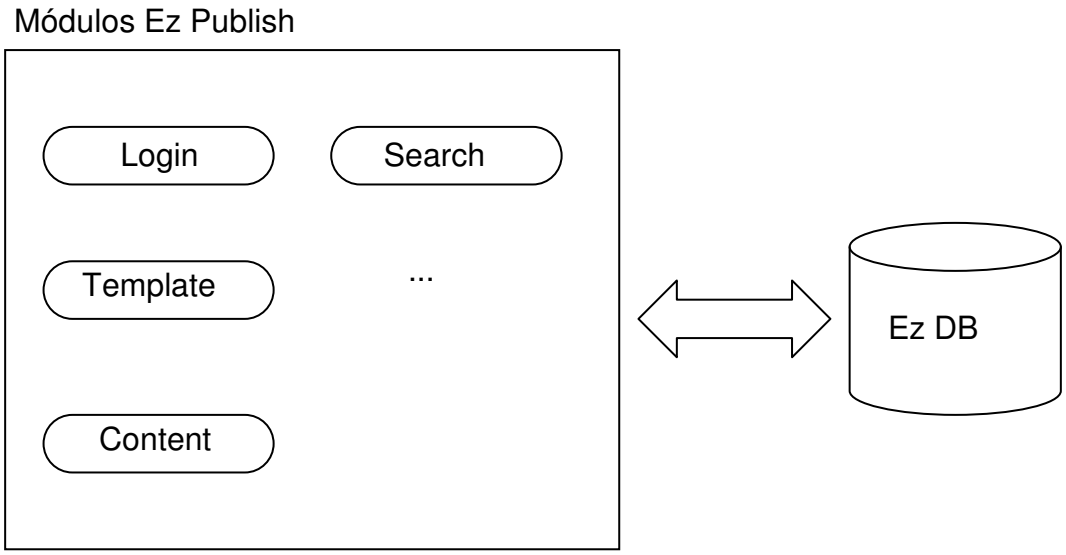

Figura 1. Módulos principales de Ez Publish

Ya que es un entorno Web todos los módulos se relacionan entre sí para dar el resultado final. En la Figura 2 puede observarse el módulo *template* en la presentación general del sitio, el módulo *content* con las categorías de documentos del sistema (parte izquierda), el módulo *login* en la parte central de la pantalla y el módulo *search* en la parte derecha, a continuación se explicará con mayor detalle cada uno de ellos.

- Módulo "*Login*": este módulo se encarga de validar los usuarios permitidos en el sistema mediante el usuario y clave asignada contra la base de datos del gestor. (Véase Figura 2)

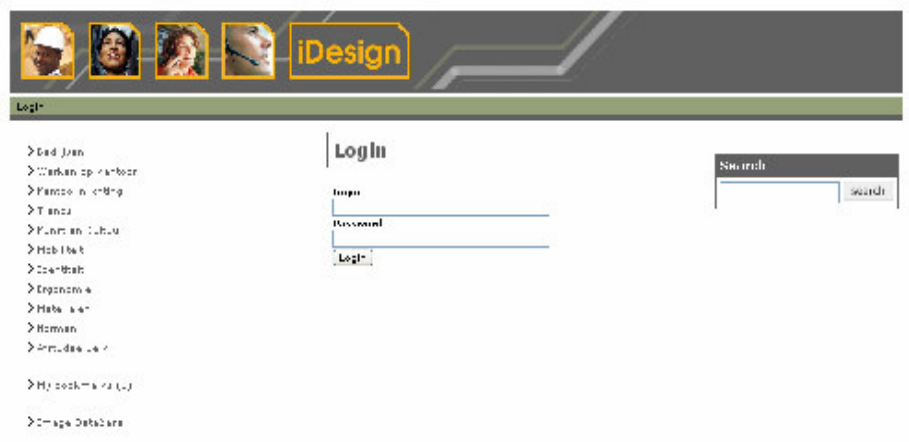

Figura 2. Página de inicio del sistema

- Módulo "*Template*": Es el módulo encargado de la presentación gráfica del sitio Web. Dentro de la jerarquía de directorios de Ez Publish se encuentra *design*, en este sitio se albergan todos los archivos (extensión tpl) encargados de dar el formato de presentación al usuario final de todo el sistema. Los archivos *tpl* son una combinación de HTML (lenguaje básico para programación de páginas Web) y funciones específicas que son interpretadas por el módulo.

- Módulo "*Content*": Este módulo se encarga de toda la gestión de contenidos dentro del sitio Web, realiza distintas consultas sobre la base de datos y envía la información al módulo *template* que se encarga de presentar los datos a los usuarios finales.

En la Figura 3 pueden observarse dos ejemplos de la forma en que ambos módulos trabajan conjuntamente.

Figura 3. Presentación del contenido por los módulos *template* y *content*

- Módulo "*Search*": este es el módulo encargado de manejar la adición, ordenamiento y búsqueda de los contenidos dentro del sistema. Interactúa directamente con la base de datos.

Figura 4. Ejemplo resultado de búsqueda original del sistema

6.2.2 Operación del módulo de adición, búsqueda y borrado de contenido

La operación de este módulo es realizada por una sub-clase llamada *ezSearchEngine*. Como puede apreciarse en el caso de uso (ver Figura 5) tiene tres funciones básicas que funcionan de la siguiente forma:

- AgregarContenido (addObject): normaliza el texto a ingresar al sistema (colocarlo en mayúsculas o minúsculas) y procede a agregar cada palabra en un arreglo con un identificador, posteriormente busca en la base de datos cuales palabras ya existen como índices y actualiza su peso[21] seguidamente añade los nuevos términos (aquellos que no existían como índices) y actualiza la base de datos
- EliminarContenido (removeObject): con el identificador del objeto busca todas las incidencias, procede a borrar los pesos de los términos y finalmente borra todas las incidencias que existan del identificador del objeto.

[21] Véase en el marco teórico el concepto de peso para las funciones de búsqueda

- BuscarContenido (search): normaliza el texto y se procede a buscar las ocurrencias en la base de datos, finalmente procesa el resultado obtenido de la búsqueda en una variable que es retornada y posteriormente interpretada por el módulo *template*.

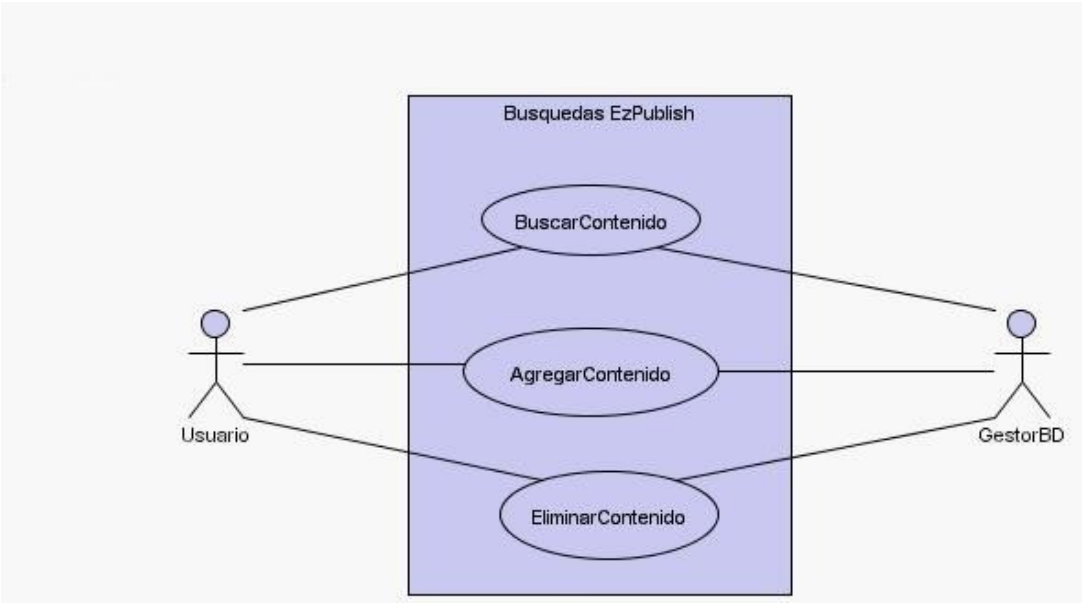

Figura 5. Diagrama de Caso de Uso para el módulo *Search* de Ez Publish

Haciendo uso de las capacidades de procesamiento y almacenamiento de los sistemas de cómputo actuales y en virtud de proporcionar resultados más acertados en los sistemas de búsquedas, se están omitiendo las distintas técnicas de pre-procesamiento de textos y se está utilizando la totalidad de los términos para construir los distintos índices para la búsqueda.

Esta tendencia es posible observarla en la forma en que el diseño del sistema Ez Publish fue construido, especialmente en la explicación de adición de contenido realizada anteriormente.

6.3 NUEVAS ADICIONES DE SOFTWARE AL SISTEMA

Aunque WordNet es desarrollado en la Universidad de Princeton, la versión utilizada para el proyecto es una modificación a la versión 1.7.1. Esta modificación fue desarrollada en el Institute for Human and Machine Cognition (IHMC) por el profesor Juan Guillermo Lalinde P. y como particularidades incluye diferentes funciones para obtener familias de palabras de distintas formas, de acuerdo a su Synset, Soundex, Hiperónimos, etc. Además, tiene acoplado un programa llamado TreeTagger[22] que se encarga de etiquetar todas las palabras que le sean entregadas y tiene la bondad que puede clasificarnos las palabras por verbos, adverbios, sustantivos, adjetivos y palabras de parada de acuerdo a una red neuronal que ha sido previamente entrenada.

Fuera de tener esta versión de WordNet, también es necesario instalar las siguientes librerías que permitirán la conexión entre los lenguajes de programación PHP (ambiente Web) y C++ (desarrollo de WordNet y TreeTagger).

Socketcc: librería de apoyo para programación de Sockets en C++ que permite la implementación de Sockets sobre TCP y UDP de manera transparente.

Pthreadcc: librería utilizada por Socketcc para el manejo de hilos (threads).

Ambas librerías fueron desarrolladas por Jason But en la Universidad de Monash en Australia[23].

---

[22] Part of Speech, http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

[23] Monash University, http://www.monash.edu.au

6.3.1 Consideraciones sobre la instalación de las nuevas aplicaciones

Es importante mencionar los elementos básicos de la configuración de algunas de las aplicaciones utilizadas para que pudieran interactuar entre si.

PHP: si se está corriendo el servidor Apache en su versión 2.0 o posteriores. La referencia al archivo “apxs” necesario por PHP debe ser especificada con su ruta de ubicación completa (algo como: /usr/local/apache2/bin/apxs). También es necesario habilitarlo para que pueda utilizar sockets (--enable-sockets) y que incluya las librerías para interactuar con MySQL (--whit-mysql)

Socketcc y Pthreadcc: estas librerías vienen con las directivas de compilación en el archivo Makefile y la guía de instalación en el archivo Readme.

WordNet: sólo es necesario copiar los archivos a una ruta específica (por ejemplo /var/wn) y agregar a las variables de entorno las siguientes instrucciones para que el sistema pueda tener las referencias a los ejecutables y librerías de la aplicación (usualmente las variables de entorno pueden ser encontradas en el archivo /etc/profile).
export LD_LIBRARY_PATH=/var/wn/lib : lugar donde se encuentran las librerías de WordNet.
export WN_DATA_DIR=/var/wn/data/ : lugar donde se encuentran los archivos de datos utilizados por WordNet.

También es necesario modificar el script IHMC-Tagger en el directorio “bin” para que el “tagger” (etiquedador) funcione adecuadamente.

Si se desea, también es posible agregar esas rutas para que sean incluidas en el path (directorios en donde busca las aplicaciones) del sistema y el sistema operativo tenga referencias de ellas.

Ez Publish: en la página de Ez Systems se encuentran ampliamente explicados los pasos a seguir para realizar la instalación del CMS tanto para sistemas operativos Windows como Linux.

6.4 PLANTEAMIENTO DE LA SOLUCIÓN

6.4.1 Factores claves del sistema actual

La lógica que maneja el gestor de contenidos puede dividirse en dos partes que se comunican entre si: la primera que tiene que ver con la interfaz de usuario (GUI Graphical User Interface) que consta de todos los formatos de presentación. Estos formatos son definidos por medio de plantillas que son reconocidas por un conjunto de clases específicas encargadas de representarlas y mostrarlas en un navegador de Internet (Browser). Estas clases también se encargan de realizar los respectivos llamados a la segunda parte, que tiene que ver con todas los programas encargados de realizar las operaciones lógicas para administrar, almacenar, buscar y editar los contenidos ingresados al sistema.

Aunque el sistema viene preconfigurado en cuanto a su apariencia, el administrador ha realizado algunas modificaciones y mejoras en la interfaz presentada al usuario, estos son los parámetros que deben tomarse en consideración de tal forma que, para los usuarios finales, la presentación de las opciones de la expansión semántica de los elementos de búsqueda sea muy intuitiva y vaya acorde a los diseños previamente establecidos. No es posible presentar el estado original del sistema porque al recibir el proyecto estos cambios ya se habían realizado.

Luego de analizar las distintas clases y plantillas utilizadas sin profundizar en las clases encargadas de interpretarlas, se encontraron los siguientes archivos que participan directamente en las búsquedas.

search.php: este archivo que se encuentra en /kernel/content/ es el encargado de realizar la llamada a la clase que realiza la manipulación de los contenidos

(adición, eliminación y búsqueda, específicamente la búsqueda) en la base de datos y retornar los resultados a la plantilla encargada de presentarlos.

ezsearchengine.php: esta clase es la encargada de realizar la adición, eliminación y búsqueda de los contenidos en la base de datos. Se encuentra en /kernel/search/ezsearchengine/

search.tpl: esta plantilla (template) es la encargada representar los resultados obtenidos desde la clase eZSearchEngine a través de la clase Search. La ruta donde se encuentra el archivo es /design/idesign/templates/content/

search_box.tpl: este template está presente continuamente en la interfaz de usuario, se puede identificar en la parte derecha de la página Web y su función consiste en llamar a la clase Search para que realice las búsquedas desde cualquier vista del sitio.

En la Figura 6 pueden observar como se observan las plantillas anteriormente descritas dentro del CMS.

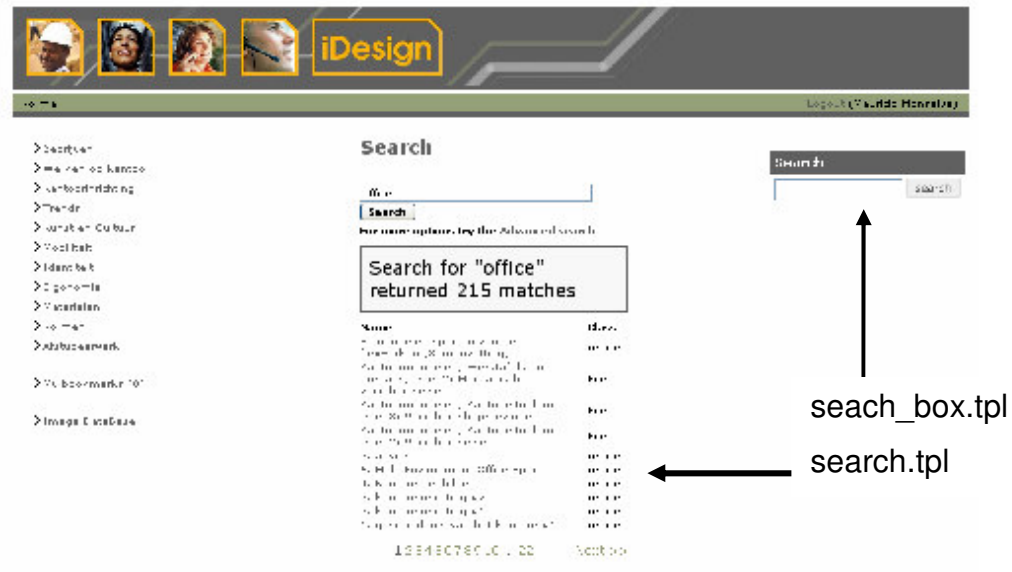

Figura 6. Ejemplo de las plantillas *search* y *search_box*

6.4.2 Criterios para sustentar la solución planteada

En las conversaciones iniciales para el desarrollo del proyecto no estaba muy claro que Ez Publish ya tenía un conjunto de herramientas elaborado para realizar la indexación de los documentos (ver anexos) y por las comunicaciones expresadas, se dió a entender que dicho mecanismo no existía y que por ende era necesario crear los algoritmos y las reglas necesarias para un proyecto de este estilo.

Al realizar la instalación de la versión 2.97 y posteriormente al hacer la actualización a la versión 3.0, se descubrió que Ez Publish poseía la estructura necesaria para realizar dichas actividades y que a pesar de su concepción en módulos los cambios de las clases dentro de las versiones tenían transformaciones considerables. Se tomó la decisión de no modificar el módulo y por ende se suprimió la necesidad de generar el mecanismo para indexar el contenido del sitio.

Otro tema importante es la manera en la cual se presentaría a los usuarios la expansión de las búsquedas en un contexto semántico. Se parte del hecho que en el sistema actualmente se tienen colecciones de 3 temas (elementos de oficina, bicicletas y jardinería), lo cual no indica que sólo se limitarán las búsquedas en estos contextos sino, que a futuro, estos temas serán más extensos y albergarán categorías diferentes.

La materia prima para la expansión semántica son las palabras con las que se realiza la búsqueda como tal, este es el punto de partida. Así que para no limitar el campo de conocimiento de los términos utilizados (necesidad del usuario) se le pedirá a WordNet que retorne de sus bases de datos todas las incidencias de los

términos para ofrecer los distintos contextos que posee como alternativas de búsqueda para los usuarios.

No se le adicionó otra carga de procesamiento al sistema como agregar a la búsqueda inmediatamente los resultados semánticos obtenidos y presentarlos dentro del mismo proceso (acción que podría considerarse invasiva por que no fue solicitada por el usuario y sin pensarlo se está sometiendo a analizar un mayor número de resultados influyendo así en demora del proceso y confusión por la diversidad de los resultados), sino que a manera de sugerencia se colocan a disposición del usuario final las opciones para que amplíe su búsqueda con nuevos términos relacionados y que probablemente no había considerado anteriormente.

6.4.3 Solución aportada

La solución propuesta consta de dos partes esenciales. La primera se encarga de añadirse al gestor de contenidos Ez Publish y la segunda es la encargada de utilizar WordNet para procesar los requerimientos de los usuarios. Cada módulo tendrá una interfaz que le permitirá comunicarse con el otro a través de IPC (Inter. Process Comunication) utilizando sockets sobre el sistema operativo.

Es evidente la necesidad de realizar modificaciones sobre las plantillas y clases que operan con Ez Publish, pero con el ánimo de no realizar cambios considerables sobre estos, se decidió crear la clase eZWordnetSearch que será llamada desde la clase Search y el resultado de su operación será entregado a la plantilla en search.tpl.

Con el fin de tener claro el concepto original de la forma en que se realiza la búsqueda obsérvese el siguiente gráfico:

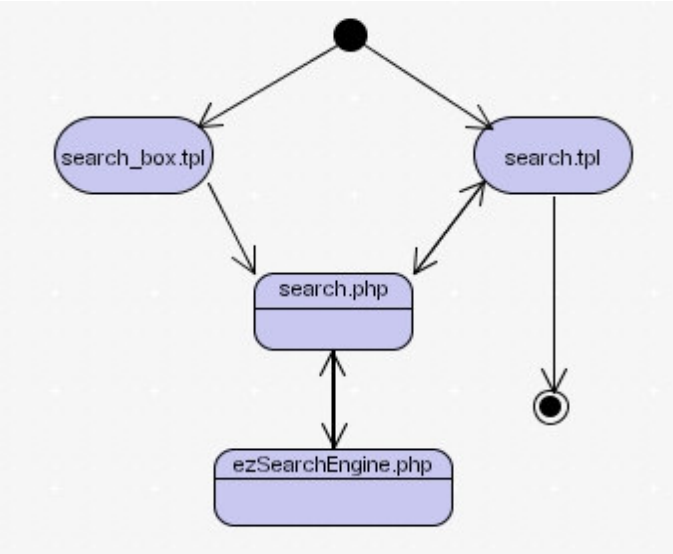

Figura 7. Esquema original del proceso de búsqueda

Como puede observarse en la figura, cualquier búsqueda puede realizarse desde dos puntos (*search_box* y *search*), cada plantilla realiza un llamado a la clase *search* quien a su vez llama a *ezSearchEngine*. Esta última clase al obtener el resultado lo retorna a la clase *search* que finalmente entrega el resultado a la plantilla *search.tpl* que finalmente realiza la presentación visual a los usuarios.

En el siguiente gráfico es posible observar como se añaden los dos módulos mencionados anteriormente que darán al sistema el complemento semántico requerido. La forma como opera la adición es la siguiente: la clase *search* llama conjuntamente a *ezSearchEngine* para buscar los resultados dentro de la base de datos de Ez como se estaba realizando y a su vez llama a *ezWordnetSearch* que utiliza la instancia creada del servidor. Este servidor se encarga de extractar los términos de búsqueda y buscarlos en WordNet que posteriormente serán presentados al los usuarios.

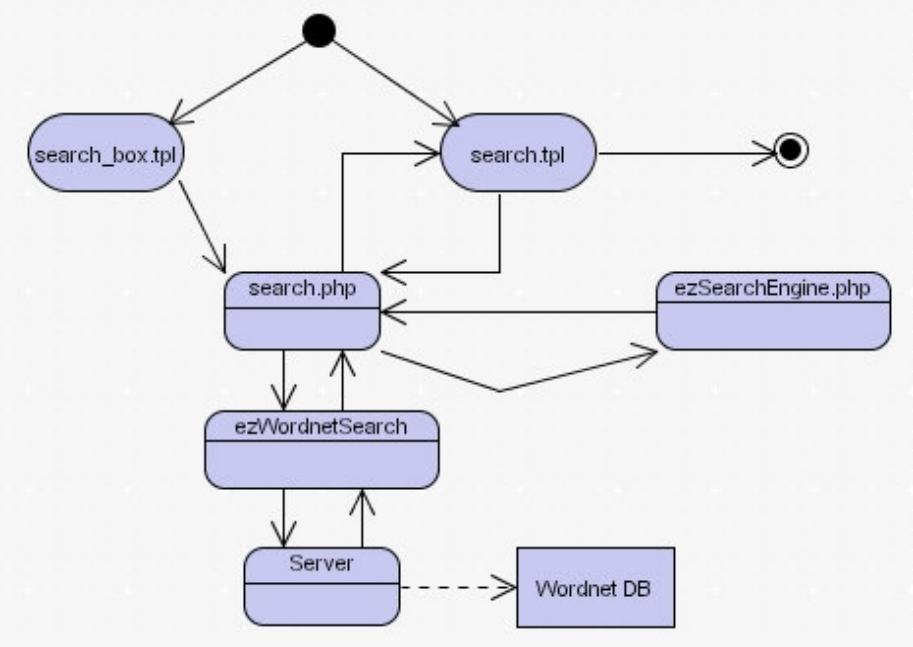

Figura 8. Nuevo esquema del proceso de búsqueda

La interacción entre la clase *ezWordnetSearch* y el servidor que interactúa con WordNet se realiza mediante sockets; el inicio de la comunicación lo establece *ezWordnetSearch* haciendo una conexión al puerto 8050 de la máquina local en donde el servidor, desarrollado en el lenguaje C++, está tomando la petición

Funcionamiento del módulo en Ez Publish

La clase eZWordnetSearch posee un método *constructor* y un método *search* que recibe como argumento el texto ingresado en la casilla de búsqueda. Este último método crea un socket que se conecta con protocolo TCP a un servidor desarrollado en C++ en el puerto 8050 de la máquina local, transmite el texto a buscar y espera la respuesta de dicho proceso. Luego recibe un flujo de caracteres que son representados por un tipo de dato “string” que es posteriormente procesado e insertado en un arreglo y este a su vez, es retornado como respuesta a la petición de búsqueda. El arreglo posee la glosa (definición)

de la palabra más el “synset” asociado a dicha glosa. Si en el flujo que recibe encuentra la cadena “ERROR” en la sección correspondiente al “synset” no agregará ese elemento al arreglo; si sólo obtuvo esta cadena, devuelve el arreglo de respuesta sólo con la glosa.

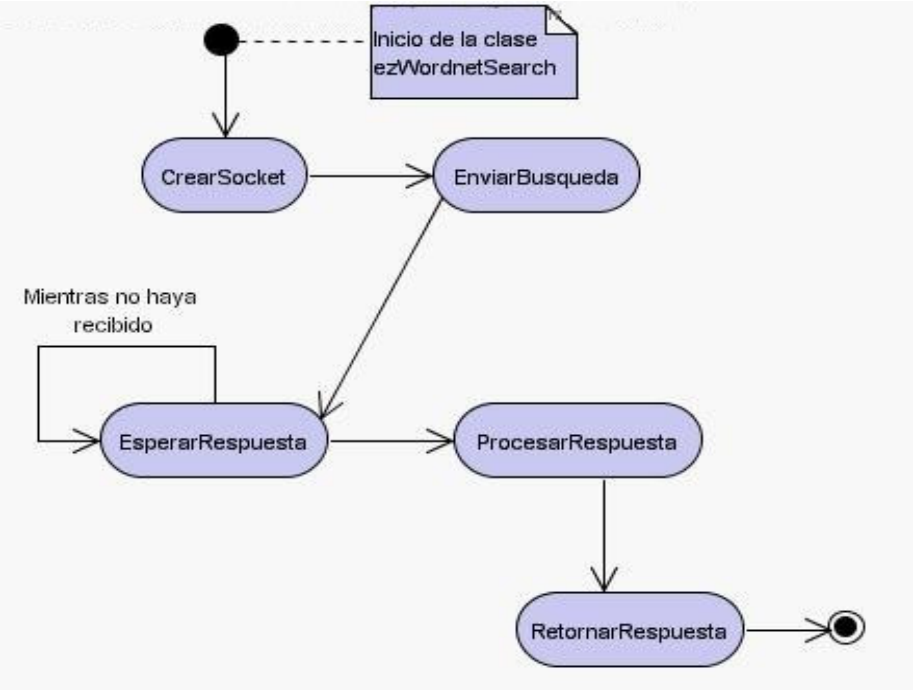

Figura 9. Diagrama de Estados para la clase *ezWordnetSearch*

En el archivo *search.php* (ver Figura 8) sólo es necesario incluir unas pocas líneas que permiten incluir el archivo *ezwordnetsearch.php* para que se tenga la referencia a la clase eZWordnetSearch, otra línea que realiza la llamada a la función *search* de la clase desarrollada y una tercera que asignará el resultado de la llamada a dos variables que serán interpretadas por la plantilla.

En la plantilla *search.tpl* (ver Figura 8), se agregan las directivas de tal forma que pueda comprender los valores que contiene el arreglo, en primer lugar presentará la glosa (significado de la palabra), y en segundo todas las palabras del SYNSET (Synonym Set, conjunto de sinónimos) en forma de enlace (URL) tal que, sólo con

un clic del ratón en el término deseado, se produzca una nueva búsqueda con dicho término como nuevo parámetro. En la siguiente figura es posible apreciar la presentación de la búsqueda antes (arriba) y después (abajo).

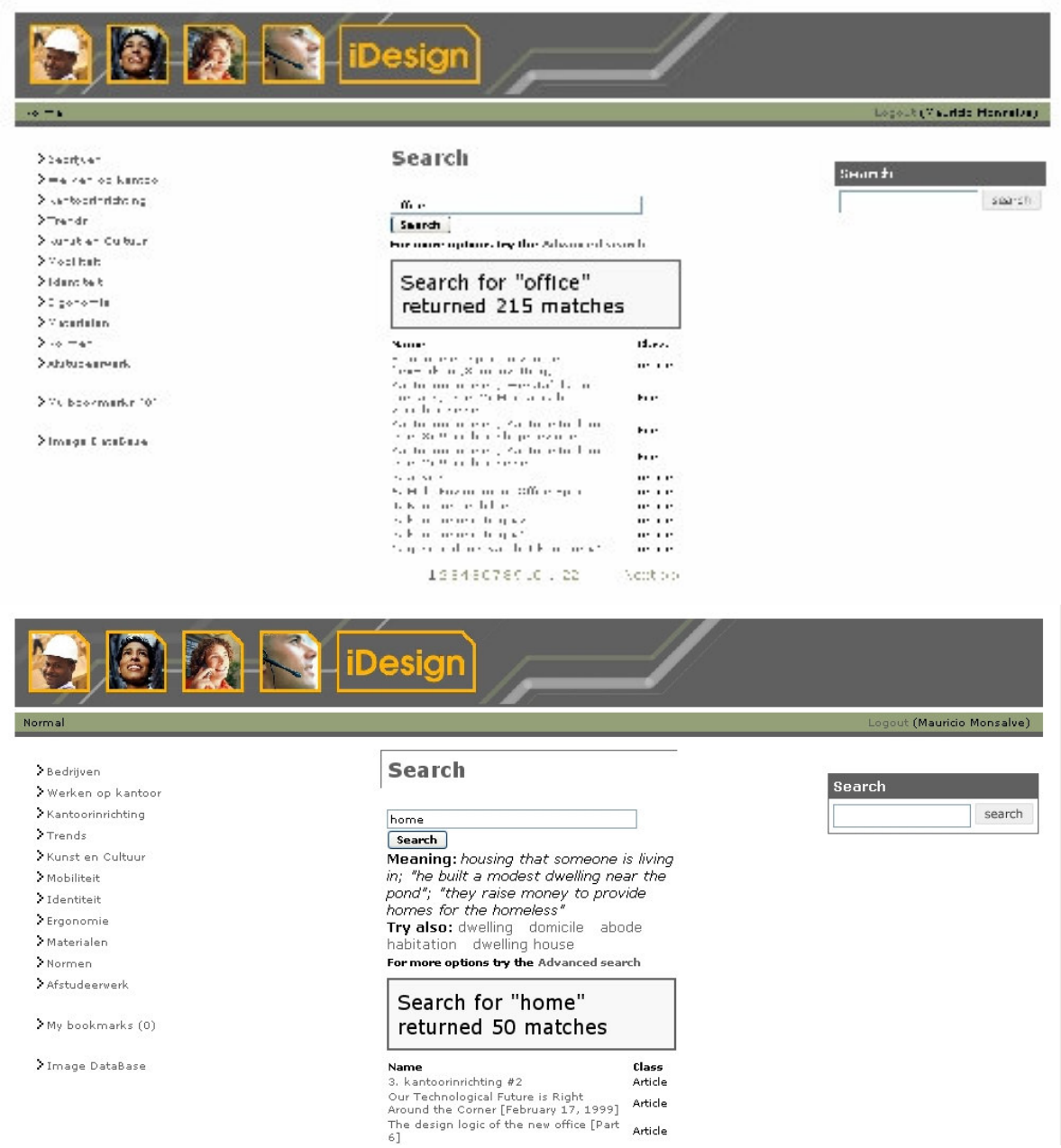

Figura 10. Nueva presentación del sitio con la mejora.

Adicionalmente en la parte inferior de la pantalla, luego de presentar todos los resultados retornados por Ez Publish, se presentarán todas las glosas retornadas por WordNet de tal forma que cada usuario que realiza la consulta tenga a su disposición los distintos contextos que posee la palabra y su respectivo conjunto de sinónimos. Al igual que el primer resultado, la información se presentará con la glosa en primer lugar seguido de los synsets asociados. En la Figura 11 se puede apreciar este caso.

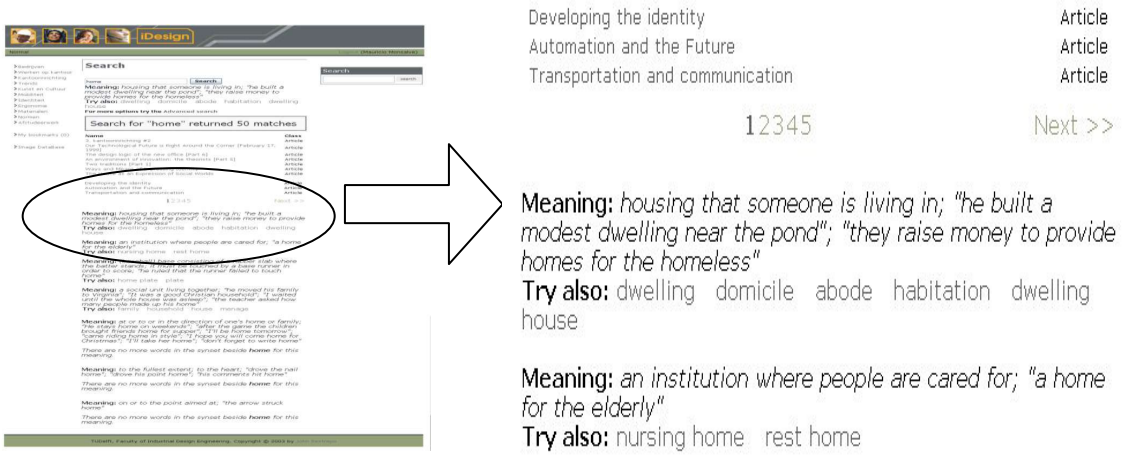

Figura 11. Resultados complementarios al final de la página

La estructura de la página Web presentada por Ez Publish es la siguiente: en primer lugar se puede apreciar la palabra "*Search*", la cual indica que es la página con los resultados de la búsqueda. Posteriormente se encuentra un cuadro de texto en el que está escrito los términos consultados. Como tercera parte se presenta el primer resultado retornado por WordNet con su definición (*Meaning*) y las palabras asociadas con este significado, si WordNet no posee más palabras en el Synset aparecerá el texto *There are no more words beside XXXXX for this meaning*, donde XXXXX es la palabra consultada.

El siguiente bloque de datos es el resultado de los documentos que Ez Publish asocia con el término de búsqueda y en la parte final de la página se encuentran

todos los resultados que fueron encontrados en WordNet presentados de la misma forma anteriormente descrita.

Funcionamiento del módulo Servidor

El proceso servidor crea la conexión al puerto 8050 de la máquina local y la instancia del objeto manejador de los archivos de la base de datos semántica *WordNet*. La función del servidor es la de estar continuamente censando el puerto, al recibir el flujo de datos, realiza el proceso de etiquetado de las palabras utilizando las funciones del *TreeTagger*. Luego de obtener el resultado de esta operación realiza un proceso de eliminación de términos o palabras de parada ubicando la operación en un nuevo vector.

Cada elemento de este nuevo vector es buscado posteriormente por la instancia de WordNet que se encarga de retornar la glosa y las palabras del Synset asociado con cada uno de los términos y finalmente retorna el resultado de la operación a través del mismo socket creado en un principio, cuando la transmisión es exitosa procede a liberar el socket para recibir mas peticiones.

Ya que la comunicación entre los dos ambientes debe realizarse a través de flujos de caracteres, también conocidos como “arreglos”, y como en WordNet algunos de los Synsets son compuestos por mas de una palabra, fue necesario colocar un caracter separador entre cada uno de los términos. Este separador es reconocido en la clase eZWordnetSearch y omitido del resultado final. También se encuentra el caracter underscore “_” que es utilizado por WordNet para separar diversas palabras que identifican un mismo elemento del Synset (términos compuestos por dos o más palabras). Estos caracteres especiales también son removidos en la clase eZWordnetSearch antes de ser ingresados al arreglo que se retorna como resultado.

6.5 MANUAL DEL USUARIO

6.5.1 Instalación

Para proceder a realizar la instalación de la adición es necesario realizar los siguientes pasos:

- Instalar *WordNet*: descomprimir el archivo *tar.gz* en el lugar de preferencia y posteriormente realizar los pasos descritos en las recomendaciones para el nuevo software[24].
- Instalar las librerías *socketcc* y *pthreadcc*
- Copiar la clase *ezWordnetSearch* que se encuentra en el archivo *ezWordnetSearch.php* en la ruta */kernel/classes* del directorio de Ez Publish
- Sobrescribir el archivo *search.php* en la ruta */kernel/content* del directorio de Ez Publish (si no se han realizado cambios en el archivo, de lo contrario es necesario copiar las líneas que están dentro de los comentarios insertados por Mauricio Monsalve)
- Sobrescribir el archivo *search.tpl* en la ruta */design/idesign/templates/content* del directorio de Ez Publish (si no se han realizado cambios en el archivo, de lo contrario es necesario copiar las líneas que están dentro de los comentarios insertados por Mauricio Monsalve)
- Copiar el servidor desarrollado en C++ en un directorio fijo desde el cual pueda ejecutarse

6.5.2 Modo de uso

[24] Numeral 6.3.1 del presente documento

Tomando en cuenta que los usuarios del sistema ya han interactuado con la versión de producción que está en estos momentos en Delft y que la idea principal fue ser lo menos agresivos posible para presentar la adición semántica, simplemente notarán una pequeña inserción de texto justo debajo de la casilla de texto en la que ingresan los términos de búsqueda que consta de dos partes. La primera identificada con "*Meaning*" en donde se encuentra la golsa, o significado, del término consultado y que proporciona el contexto en el que se encuentra. Y la segunda se identifica con el texto "*Try also:*" seguido del conjunto de palabras retornadas por la nueva clase y que corresponden al conjunto de sinónimos asociados a la palabra de consulta primaria que están relacionados con ella.

Con la intención de ser práctico y de agilizar el proceso de consulta, los términos se han colocado como enlaces (links, URL) que al ser presionados generarán una nueva búsqueda. Visualmente son fácilmente identificables al poseer las propiedades de formato que son definidas por los archivos CCS (Common Cascade Stylesheet) y acoplarse al formato definido allí para los enlaces.

El usuario que considere que alguno de los términos es útil para definir mejor su necesidad de búsqueda simplemente puede hacer un clic sobre este e inmediatamente el sistema buscará con este término como nuevo parámetro. La operación en los elementos al final de cada página Web operan de la misma forma. Véase Figura 12.

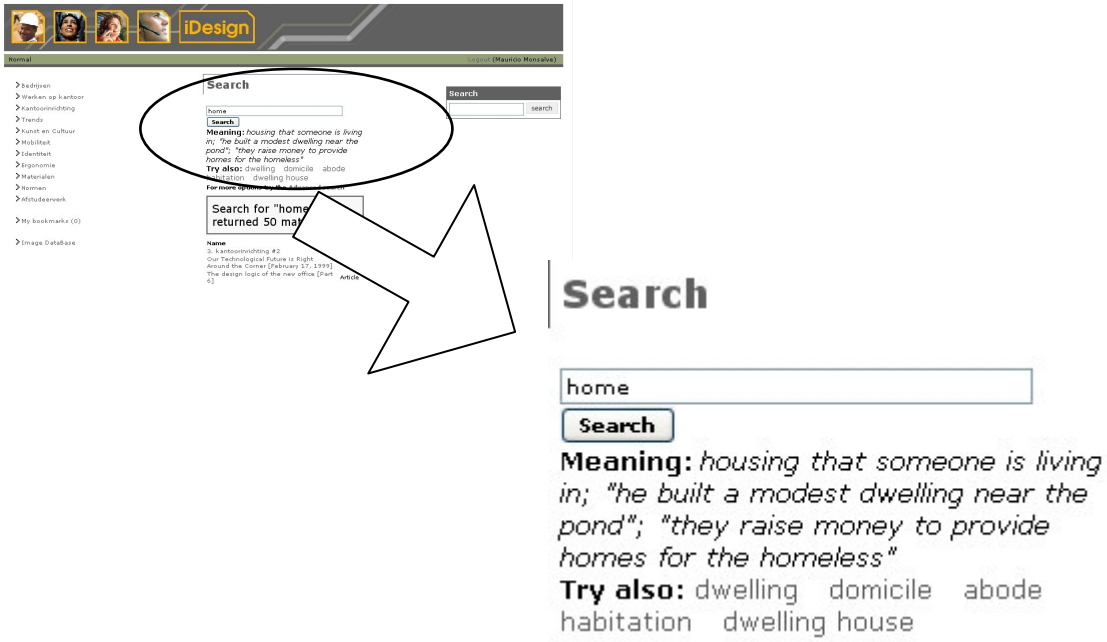

Figura 12 Presentación de la adición semántica al *template* de búsquedas

A pesar de ser tan intuitivo podría ser útil que el administrador del sistema creara un artículo en la página de inicio del sistema con un pequeño instructivo en Holandés de los cambios que se realizaron en la plantilla, la forma como los visualizarán y aclarar que los hipervínculos que se presentan debajo de la casilla de búsqueda llevarán al sistema a realizar una nueva búsqueda. Además de aclarar que todos los resultados obtenidos son en el idioma Inglés.

6.6 PROBLEMAS, EVENTUALIDADES Y SOLUCIONES DURANTE EL DESARROLLO DEL PROYECTO

Durante el desarrollo del proyecto, algunos percances impidieron el normal desarrollo de acuerdo al cronograma propuesto. En especial el que tiene que ver con la estabilidad de la máquina en la que se comenzó a realizar el desarrollo y por esta razón, fue necesario realizar la instalación completa del sistema en tres oportunidades. Esto no quiere decir que el software utilizado fuera el causante del problema; en su mayoría las especificaciones de la máquina no fueron las más adecuadas para los requisitos necesarios del sistema.

Una primera instalación inicial en donde se enfrentó los inconvenientes generales que se tienen al realizar instalaciones de aplicaciones que por no ser muy comunes y no tener un conocimiento profundo de ellas, llevaron un mayor tiempo con respecto a las posteriores. En esta primera instalación trabajaron independientemente el entorno de Ez Publish, servidor Web, PHP y MySQL; y por el otro lado el que tiene que ver con WordNet.

La segunda oportunidad se presentó cuando en el laboratorio en donde se encuentra la máquina asignada, decidieron cambiar sin realizar consulta adecuada de las implicaciones, el motor de base de datos MySQL, este hecho provocó que el ambiente del CMS dejara de funcionar y no fuera posible tener acceso a la colección de documentos. Tratando de revertir los cambios se presentó un error grave en el módulo del kernel del sistema operativo (núcleo del sistema) que manejaba el entorno gráfico y se tomó la decisión de reinstalar el sistema. Para esta ocasión trató de utilizarse el sistema operativo Fedora ya que las versiones de Red Hat con licencia GPL ya no estaban disponibles, esta vez no fue posible instalar el sistema completamente, ya que el CMS no reconocía la base de datos, el problema fue relacionado con la información que el gestor de contenidos

esperaba encontrar en la base de datos y que dicha información no estaba en la base de datos.

Para explicar el problema se debe entender que cuando se realiza una nueva instalación del sistema, este genera unos archivos que utiliza para guardar los valores de configuración que son consultados cada vez que se realiza un llamado a la página Web. Como esa instalación no era nueva sino que era un backup anterior todos estos archivos de configuración ya estaban creados, en este caso el problema se resuelve volviendo a repoblar la base de datos asignándole el mismo nombre que se encuentra en los archivos de configuración.

Teniendo en cuenta este inconveniente, que en ese momento no se conocía la solución y puesto que la parte conceptual del campo de recuperación de información era muy extensa y que sería necesario dedicarle un tiempo considerable al análisis de esta información, se presentó una ausencia del campus de algunos meses.

Al volver a retomar el trabajo se informó que el equipo de cómputo había sufrido una falla grave al realizársele un mantenimiento correctivo y que por este motivo ya no funcionaba, allí fue la tercera oportunidad, esta vez fue necesario buscar una nueva máquina y proceder a la instalación de los componentes nuevamente unas pocas semanas antes de la fecha límite para la entrega del proyecto. Se refuerza que la máquina que fue asignada al proyecto era inestable de por si y no era la adecuada para el desarrollo del mismo.

Durante el transcurso del tiempo del proyecto fue necesaria la intervención del asesor Juan Guillermo Lalinde y el coordinador de proyectos Hernán Toro para resolver algunos problemas con las políticas de seguridad establecidas por el área de telemática del centro de informática de la Universidad que impidieron que el asesor y persona de contacto en Holanda pudiera brindar la asesoría adecuada al

no poder tener acceso a la información en el equipo de cómputo, implicando un retraso en el cronograma de trabajo.

Las posibilidades de aprendizaje y el análisis de gestor de contenidos sufrió una demora considerable al presentarse un cambio de versión en el CMS y además de esto una reestructuración del sitio Web de la compañía Ez Systems en la que todas las referencias a la documentación del sistema dejaron de estar disponibles, impidiendo resolver las dudas que se presentaron y que a su vez John Restrepo, el asesor en Holanda, desconocía por haber cambiado en la nueva versión del sistema o por no tener la necesidad de utilizarlas anteriormente.

Con respecto a los objetivos planteados en el anteproyecto de forma inicial, dos de ellos no se cumplieron por los siguientes motivos.

El de diseñar un mecanismo de indexación para los documentos y su posterior inserción a la base de datos no fue necesario de desarrollar, ya que, como se explicó en el marco teórico acerca de los sistemas de recuperación de información, para los sistemas actuales no es necesario utilizar ningún tipo de manipulación sobre los textos, además que este cambio significaría una reforma completa al módulo de búsqueda actual. Además luego de realizar el análisis respectivo del módulo habilitado en Ez Publish para esto, se llegó a la conclusión que la forma en que este trabaja es adecuada y apropiada para las necesidades de los investigadores en Delft.

El hecho de cambiar el sistema de búsqueda, por ejemplo cambiando los algoritmos de forma que los resultados presentados incluyan en la búsqueda documentos relevantes al conjunto de Synsets asociados al término inicialmente solicitado conjuntamente con los resultados individuales de la palabra enfrentaría al usuario del sistema a evaluar una mayor cantidad de documentos, logrando así que el tiempo invertido en consultar información se aumentara considerablemente

y que posiblemente este mecanismo no fuera óptimo para sus necesidades ya que en vez de proporcionársele información detallada y precisa, se le está presentando gran cantidad de información posiblemente no relevante y que por estar allí podría desviarlo de su objetivo inicial.

Con respecto a crear una opción de personalizar el sistema de tal forma que delimitara las opciones de búsqueda de acuerdo a los elementos más frecuentes en la interacción con los usuarios, esta parte del proyecto se cimentó sobre una mejora realizada a WordNet realizada en la versión 1.5, lastimosamente la estructura de los archivos de datos utilizados por la aplicación poseen una alta complejidad ya que sus referencias de identificación corresponden a posiciones estáticas de la ubicación de la información dentro del archivo. Por esta razón no fue posible adicionar el campo de conocimiento asociado a los archivos en la versión 1.7 utilizada durante el proyecto que serían el principio para realizar el manejo personalizado de las búsquedas de acuerdo a los temas de interés de los usuarios.

7 CONCLUSIONES

7.1 SOBRE EZ PUBLISH

En el transcurso del proyecto fue posible conocer tres versiones de este CMS y emitiendo un concepto desde la perspectiva por la cual se analizaron, es un sistema que es robusto, fácil de manejar y administrar para sitios de poca complejidad. Al momento de realizar cambios sobre el sistema, personalizarlo tanto en apariencia como en funcionalidades, es posible realizarlo aunque la curva de aprendizaje para manejar todas las funciones proporcionadas puede ser un poco lenta ya que la documentación disponible no se encuentra bien estructurada especialmente en el tema de las plantillas (*templates)*.

Si las necesidades involucran el desarrollo de un nuevo módulo el CMS aporta un mecanismo para incluir extensiones. Esta funcionalidad se abordó con el fin de considerarla como el mecanismo óptimo para adicionar el módulo que conectaría el sistema con WordNet pero no fue posible adaptarla de esta manera, por tanto se tomó la decisión de añadir unas líneas en el código que Ez Systems proporciona.

Dada la necesidad de modificar el código proporcionado, se observó que es sencillo realizar el seguimiento de las plantillas y clases utilizadas una vez se posee un conocimiento básico sobre el sistema. Además el desarrollo de nuevos módulos, al ser programados en el lenguaje PHP (similar a C++ y Java) simplifica la complejidad para los desarrolladores que han trabajado con dichos lenguajes o para nuevos programadores que comienzan su estudio por lenguajes orientados a objetos.

En casos que sea necesario programar algún código que no sea posible realizarlo desde una parte distinta al código fuente de Ez Publish, se recomienda que estrictamente ese cambio siempre sea utilizado con la misma versión del CMS, esto se debe a que los cambios son bastante considerables entre versiones, y de esta forma pueda incluirse en múltiples sistemas que utilicen la misma versión.

7.2 SOBRE WORDNET

Esta es una poderosa base de datos semántica, con las pruebas realizadas posee una gran variedad de conceptos y en especial para el contexto del presente proyecto presenta resultados muy apropiados y diversos a pesar que sólo se aprovechó una pequeña parte de las funciones que fueron desarrolladas en el *Institute for Human and Machine Cognition* (IHMC). Es importante que la versión original del laboratorio de la Universidad de Pincteon también puede ser utilizada sin problemas; es necesario realizar algunos cambios sobre el servidor desarrollado en C++ para que incluya la respectiva referencia a las clases y funciones correctas para esta versión.

Existe también un trabajo importante que fue realizado para la versión 1.5 que consiste en agregar al final de cada synset el contexto al que este pertenece, de esta forma es posible obtener mayores asociaciones con familias de palabras que están relacionadas dentro de dicho contexto. En este proyecto no fue posible utilizar esta opción porque la estructura como están conformados físicamente los archivos que constituyen a WordNet es muy estricta y cualquier modificación sobre estos implica un cambio general de todo el archivo.

Aparte de estos trabajos ya mencionados también resaltamos uno en especial que fue desarrollado con el fin de utilizar todos los archivos de WordNet directamente con el lenguaje PHP. La ventaja que puede presentar este sistema es que quienes

no están muy acostumbrados a trabajar con C++ pueden realizar las llamadas desde su sitio desarrollado con PHP e incluir los resultados gracias a esta interfaz[25].

Para utilizar esta herramienta en lugares donde el idioma inglés no es la mejor opción, existe la posibilidad de hacer uso de EuroWordnet[26], esta pretende dar soporte para los idiomas Holandés, Italiano, Español, Alemán, Francés, Checo y Estoniano de la misma forma que lo hace la versión original de WordNet.

7.3 SOBRE EL PROYECTO

Existe un factor importante a considerar y es el hecho que el proyecto se trabaje a "control remoto" (Como se describe en uno de los mensajes que se encuentran en el Anexo A) lo cual implica dificultad en la comunicación adecuada de la persona que conocía el entorno de Ez Publish y debido a esto algunos retrasos que se presentaron debido al conocimiento incipiente de este sistema hubiesen sido resueltos más rápidamente al encontrarse en distancias más cercanas (por ejemplo la misma ciudad, o la misma zona horaria).

Es importante aclararle a cualquier persona que desee continuar con el presente proyecto o establecer uno nuevo a partir de este, que se dedique a conocer el entorno de Ez Publish. Sería interesante establecer una agenda con temas específicos que puedan acordarse con Jhon Restrepo (encargado del sitio en Holanda) para que el resto del proyecto no presente dificultades en este tema.

---

[25] Este proyecto fue desarrollado por Mike McDonald, http://www.foxsurfer.com/wordnet/

[26] Universidad de Amsterdam http://www.illc.uva.nl/EuroWordNet/

El proyecto posee un gran potencial que puede aplicarse a las distintas formas en que WordNet puede proporcionar los resultados frente a las búsquedas (consultar como verbos, sustantivos, adverbios, etc.) simplificando de esta forma el universo de resultados posibles para cada consulta.

8 TRABAJO FUTURO

Se sugieren como temas propuestos para continuar con el desarrollo y mejora del actual proyecto:

Convertir los archivos de la versión 1.5 de WordNet a la versión 1.7, esta conversión permitirá tener un campo de conocimiento asociado a cada synset y con esta opción es posible crear una nueva tabla en la base de datos que almacene los temas de consulta más buscados por los usuarios de tal forma que las adiciones semánticas presentadas en un futuro puedan reducirse a determinados campos del interés de las personas que interactúan con el sistema.

Es también interesante considerar la posibilidad de indicarle a WordNet que tipo de palabras se desean visualizar como resultado de su operación. De este modo, el usuario tendría la posibilidad de ingresar fuera del texto a buscar una opción que le permita escoger entre las relaciones semánticas y léxicas como sinónimos (el alcance de este proyecto), antónimos, hiperónimos, hipónimos, etc. con el fin de ampliar de otro modo el marco de búsquedas.

Realizar un desarrollo de un CMS, puede ser Ez Publish, para albergar una colección de documentos en español e integrarle EuroWordNet, que es la versión de WordNet multi-lenguaje y así prestar funcionalidad para la comunidad hispana. Es posible que un sistema de este estilo pueda ser útil en la Universidad EAFIT para el manejo de las producciones culturales e intelectuales de sus empleados de las distintas escuelas o por el Fondo Editorial para los resúmenes de obras publicadas.

9 ANEXOS

9.1 Anexo A

Correos electrónicos

Mensaje 1
De: Juan Guillermo Lalinde [jlalinde@eafit.edu.co]
Enviado: Miércoles, 05 de Febrero de 2003 03:54 p.m.
Para: mmonsal1@eafit.edu.co
Asunto: Proyectos de Grado Sistemas (fwd)

--

JUAN GUILLERMO LALINDE PULIDO
Universidad EAFIT
E-mail: jlalinde@sigma.eafit.edu.co

---------- Forwarded message ----------
Date: Wed, 29 Jan 2003 17:50:59 +0100
From: John Restrepo <j.restrepo@io.tudelft.nl>
To: Alberto Rodríguez <arodrig@sigma.eafit.edu.co>,
Juan G. Lalinde <jlalinde@ai.uwf.edu>, jlalinde@eafit.edu.co
Subject: Proyectos de Grado Sistemas

Juan (y Alberto)

Discúlpame que me haya demorado tanto en responderte y en aparecer con ideas sobre cómo podemos trabajar juntos. Creo que mediante proyectos de grado de estudiantes podemos hacer cosas juntos. Crees que podemos tener algunos estudiantes de sistemas haciendo proyecto de grado a control remoto? Tengo dos temas en los que trabajar, de los que ya habíamos hablado, y en los que hay intereses comunes. Un profe de Eafit puede ser co-director. En el segundo tema creo que podrías ser tú mismo, no? Los títulos son provisionales.

Me cuentas que piensas. Si estás de acuerdo, podemos empezar lo más rápido posible.

Saludos,

John Restrepo

Bases de datos multimedia. Query by example.
En este proyecto se desarrollará un cliente para hacer consultas en una base de datos multimedia mediante ejemplos. La consulta se hace dándole a la base de datos una imagen (o una colección de imágenes). Las consultas se hacen usando MRML (una aplicación de XML a bases de datos multimediales)

El proyecto consiste en:
- Desarrollar un cliente para consultar la base de datos
- Agregar al motor algunas funciones como múltiples sesiones, búsquedas simultáneas en diferentes colecciones, e indexado 'en vivo'
- El cliente será desarrollado usando un SDK existente y será integrado a un Sistema de Gestión de Información también existente

El candidato tendrá acceso a:

- El software necesario (todo freeware/shareware para uso académico)
- Acceso SSH a un servidor en Delft, donde estará alojado el proyecto y dónde se harán las pruebas
- Acceso a las colecciones de imágenes con las que se trabaja
- Literatura relevante
- Instrucción y retroalimentación del responsable en Delft.

Se requieren:
- Conocimientos en PHP, Java y XML Indispensables
- Conocimientos en Perl son deseables.
- El estudiante requiere acceso a una máquina Linux en Medellín donde pueda hacer sus desarrollos.

Implementación de WordNet en un Sistema de Gestión de Información En este proyecto se implementará un sistema de recuperación de información usando WordNet. Los resultados se presentarán de manera gráfica (ver por ejemplo www.kartoo.com). Existe la posibilidad de que las consultas también se hagan de forma gráfica. Los requisitos del sistema son:

- Debe usar el SDK del Sistema de Gestión de Información existente
- Las búsquedas están basadas en lo que el sistema sabe sobre el usuario y/o sobre el proyecto en el que esta trabajando.
- Proponer, de manera proactiva información que no ha sido específicamente requerida por un usuario, basándose en lo que el sistema sabe del usuario, del proyecto en el que está trabajando, y en lo que otros usuarios han encontrado útil.

El candidato tendrá acceso a:

- El software necesario (todo freeware/shareware para uso académico)
- Acceso SSH a un servidor en Delft, donde estará alojado el proyecto y dónde se harán las pruebas
- Acceso a las colecciones de imágenes con las que se trabaja

- Literatura relevante
- Instrucción y retroalimentación del responsable en Delft.

Se requieren:
- Conocimientos en PHP, Java y XML Indispensables
- Conocimientos en Perl son deseables.
- El estudiante requiere acceso a una máquina Linux en Medellín donde pueda hacer sus desarrollos.

John Restrepo <j.restrepo@io.tudelft.nl>
TU Delft
Industrieel Ontwerpen
Landbergstraat 15 Kamer 10-4B-39
2628CE Delft
The Netherlands
Phone +31 15 278-5146 Fax +31 15 278-7179

Mensaje 2
De: Juan Guillermo Lalinde [jlalinde@eafit.edu.co]
Enviado: Viernes, 21 de Febrero de 2003 11:14 a.m.
Para: mmonsal1@eafit.edu.co
Asunto: Información de Holanda: EzPublish3 (fwd)

--

JUAN GUILLERMO LALINDE PULIDO
Universidad EAFIT
E-mail: jlalinde@sigma.eafit.edu.co

---------- Forwarded message ----------
Date: Wed, 19 Feb 2003 12:03:11 +0100
From: John Restrepo <j.restrepo@io.tudelft.nl>
To: Juan G. Lalinde <jlalinde@ai.uwf.edu>
Subject: EzPublish3

Juan,

Acabo de instalar eZPublish 3 y se ve super bueno. el SDK y los manuales

los puedes ver en línea una vez tengas instalado el sw. Hay versiones para
win, mac y linux (yo uso la de linux por razones obvias).
Estoy trabajando en la migración de la base de datos de la versión anterior
a la nueva. Creo que eso me tomará un par de días.

Cómo propones que empecemos a trabajar con tu estudiante? Cómo vamos a escribir la definición de su proyecto? Podemos conseguir una cuenta en Linux en la U para él (con shell access)? Cómo nos vamos a comunicar (chat,
e-mail, etc?)

Saludos,

_____
John Restrepo <j.restrepo@io.tudelft.nl>
TU Delft
Industrieel Ontwerpen
Landbergstraat 15 Kamer 10-4B-39
2628CE Delft
The Netherlands
Phone +31 15 278-5146 Fax +31 15 278-7179
_____

Mensaje 3
De: Juan Guillermo Lalinde [jlalinde@eafit.edu.co]
Enviado: Viernes, 21 de Febrero de 2003 11:15 a.m.
Para: mmonsal1@eafit.edu.co
Asunto: Re: Proyectos de Grado Sistemas (fwd)

--

JUAN GUILLERMO LALINDE PULIDO
Universidad EAFIT
E-mail: jlalinde@sigma.eafit.edu.co

---------- Forwarded message ----------
Date: Mon, 10 Feb 2003 17:16:02 +0100
From: John Restrepo <j.restrepo@io.tudelft.nl>

To: Juan Guillermo Lalinde <jlalinde@eafit.edu.co>
Cc: Alberto Rodríguez <arodrig@sigma.eafit.edu.co>

Subject: Re: Proyectos de Grado Sistemas

Juan,

Buenas noticias, Gracias.

el SDK de EzPublish está en internet. La dirección es: http://sdk.ez.no
Estoy esperando la versión 3. Actualmente uso la 2.2.9 pero la 3.0 es una generación completamente nueva. De la versión 3 ya hay betas estables, así
que se puede empezar a trabajar.

Una descripción se puede ver en http://www.ez.no/article/view/398 una introducción general buenísima se puede bajar de
http://www.ez.no/filemanager/download/477/eZ_publish_3_Presentation.pdf
Los manuales de la versión 3 se pueden ver en http://194.248.150.24/

Leeré las fuentes que me enviaste y te cuento.

Çiao,

John

John Restrepo <j.restrepo@io.tudelft.nl>
TU Delft
Industrieel Ontwerpen
Landbergstraat 15 Kamer 10-4B-39
2628CE Delft
The Netherlands
Phone +31 15 278-5146 Fax +31 15 278-7179

Mensaje 4
De: Juan Guillermo Lalinde [jlalinde@eafit.edu.co]
Enviado: Viernes, 21 de Febrero de 2003 11:43 a.m.
Para: John Restrepo
CC: mmonsal1@eafit.edu.co
Asunto: Re: EzPublish3

John.
Acabo de enviarle la información al estudiante para que la mire. Ya está trabajando con WordNet. Creo que un mecanismo interesante puede ser el chat. Por el momento, el estudiante se llama Mauricio Monsalve y el email es mmonsal1@eafit.edu.co. Le envío copia del mensaje para que te puedas poner en contacto con él directamente si lo necesitas.

Juan Guillermo

On Wed, 19 Feb 2003, John Restrepo wrote:

>
> Juan,
>
> Acabo de instalar eZPublish 3 y se ve super bueno. el SDK y los
> manuales
> los puedes ver en línea una vez tengas instalado el sw. Hay versiones
para
> win, mac y linux (yo uso la de linux por razones obvias).
> Estoy trabajando en la migración de la base de datos de la versión
anterior
> a la nueva. Creo que eso me tomará un par de días.
>
> Cómo propones que empecemos a trabajar con tu estudiante? Cómo vamos a
> escribir la definición de su proyecto? Podemos conseguir una cuenta en
> Linux en la U para él (con shell access)? Cómo nos vamos a comunicar
(chat,
> e-mail, etc?)
>
> Saludos,
>
>
>
> _____
> John Restrepo <j.restrepo@io.tudelft.nl>
> TU Delft
> Industrieel Ontwerpen
> Landbergstraat 15 Kamer 10-4B-39
> 2628CE Delft
> The Netherlands
> Phone +31 15 278-5146 Fax +31 15 278-7179
> _____
>
>
>

--

JUAN GUILLERMO LALINDE PULIDO
Universidad EAFIT
E-mail: jlalinde@sigma.eafit.edu.co

Mensaje 5
De: John Restrepo [j.restrepo@io.tudelft.nl]
Enviado: Lunes, 03 de Marzo de 2003 05:08 a.m.
Para: Mauricio Monsalve; Juan G. Lalinde
Asunto: Re: Proyecto de grado. WordNet y EZPublish

Hola Mauricio,

Gusto 'conocerte'. (Esta era de internet si nos hace decir unas cosas...)

Gracias por interesarte en este proyecto. Hay algunas cosas que quería decirte, aquí van:

a.. En el sitio de EzPublish tienen unas versiones del CMS 'instalables'. Estas versiones vienen con apache, php y Ez. Las hay para windows y para Linux. Mi sugerencia es que NO uses las versiones instalables, pues son muy inflexibles.
b.. La instalación de EzPublish es muy simple pero no todo lo que hay que saber está en la documentación. Aquí varios tips:
    a.. Aunque se puede trabajar en windows y en linux, yo preferiría que trabajásemos en Linux (Juan Guillermo, estás de acuerdo?)
    b.. Baja la versión de EzPublish normal (actualmente tienen una versión que es la 2.9.7, que es la beta 2 de la nueva versión 3.) Puedes usar esa. La encuentras en http://www.ez.no/filemanager/download/538/ezpublish-2.9-7.tar.gz
    c.. Necesitas PHP versión >= 4.1.x
    d.. Apache 1.3 (Recomendado) pero a mi me funciona bien con apache 2
    e.. La base de datos que uso es MySQL
    f.. Es necesario aumentar la memoria máxima que un script puede usar de 8 a 12 Mb (en php.ini)
    g.. Ellos sólo dan instrucciones de cómo instalar usando virtual hosting cambiando el nombre del host para los diferentes sitios. En mi caso esto no es posible porque no tengo acceso al DNS, así que lo resolví usando puertos. Agregar lo siguiente al archivo de configuración de apache (cambiar IP's, nombres, puertos, etc. según sea necesario, este es un ejemplo de mi archivo de configuración)

```
      LoadModule rewrite_module modules/mod_rewrite.so


      Listen 130.161.172.229:3300
      Listen 130.161.172.229:3301
      Listen 130.161.172.229:3302

      <VirtualHost 130.161.172.229:3300>
       <Directory /var/www/ezpublish-2.9-7/>
             Options FollowSymLinks Indexes ExecCGI
             AllowOverride None
        </Directory>

        RewriteEngine On
        RewriteRule !\.(gif|css|jpg|png)$ /var/www/ezpublish-2.9-
7/index.php

        ServerAdmin j.restrepo@io.tudelft.nl
        DocumentRoot /var/www/ezpublish-2.9-7/
        ServerName dutodj.io.tudelft.nl
      </VirtualHost>
```

```
      <VirtualHost 130.161.172.229:3301>
       <Directory /var/www/ezpublish-2.9-7/>
             Options FollowSymLinks Indexes ExecCGI
             AllowOverride None
        </Directory>

        RewriteEngine On
        RewriteRule !\.(gif|css|jpg|png)$ /var/www/ezpublish-2.9-
7/index.php

        ServerAdmin j.restrepo@io.tudelft.nl
        DocumentRoot /var/www/ezpublish-2.9-7/
        ServerName dutodj.io.tudelft.nl
      </VirtualHost>

      <VirtualHost 130.161.172.229:3302>
       <Directory /var/www/ezpublish-2.9-7/>
             Options FollowSymLinks Indexes ExecCGI
             AllowOverride None
        </Diectory>

        RewriteEngine On
        RewriteRule !\.(gif|css|jpg|png)$ /var/www/ezpublish-2.9-
7/index.php

        ServerAdmin j.restrepo@io.tudelft.nl
        DocumentRoot /var/www/ezpublsh-2.9-7/
        ServerName dutodj.io.tudelft.nl
      </VirtualHost>
    h.. Luego cambiar en
directorio/donde/instalé/ezpublish/settings/site.ini
      [PortAccessSettings]
      # Add entries here if you have port in MatchOrder
      # Each entry consists of the port=accessname
      3300=user
      3301=admin
      3302=demo
    i.. Hay que generar la documentación de la API. Para esto corre el
script que hay en directorio/donde/instalé/ezpublish/bin/shell/makedoc.sh
    j.. Una vez instalado, puedes acceder al manual y al SDK en tú
máquina. Estarán en:
      a.. http://mi.servidor.eafit.edu.co:3301/manual
      b.. http://mi.servidor.eafit.edu.co:3300/sdk
    k.. Ésto, junto con el resto de las instrucciones que aparecen en la
documentación debe ser suficiente.
    l.. Notarás que la documentación de las API Reference no está lista
en esta versión del SDK (Al hacer click allí aparece una página en
blanco). Hay una solución...
      a.. Baja ezphpdoc de
http://www.ez.no/filemanager/download/3/ezphpdoc-1.0.tar.gz
      b.. corre el script phpdoc-1.0.pl directorio/de/ezpublish/kernel -o
directorio/donde/quiero/la/documentación
      c.. Listo, mira
directorio/donde/quiero/la/documentación/html/index.html
```

Si tienes problemas me cuentas.

Saludos,

John Restrepo

_____
John Restrepo <j.restrepo@io.tudelft.nl>
TU Delft
Industrieel Ontwerpen
Landbergstraat 15 Kamer 10-4B-39
2628CE Delft
The Netherlands
Phone +31 15 278-5146 Fax +31 15 278-7179
_____

Mensaje 6
De: John Restrepo [j.restrepo@io.tudelft.nl]
Enviado: Jueves, 06 de Marzo de 2003 03:40 a.m.
Para: Mauricio Monsalve; Juan G. Lalinde
Asunto: Re: Pregunta

Mauricio,

>1. Cuanto sera el numero de personas que estaran utilizando el
>proyecto? 2. Ya han intentado establecer cuantas consultas (aprox.) se
>realizan por dia?

El trabajo que estoy haciendo tiene como objetivo estudiar cómo los diseñadores interactúan con fuentes de información, en especial, aquellas que se accesan a través de un computador (no libros, colegas, etc. que también pueden ser fuentes de información)

Para demostrar lo que se ha aprendido en experimentos en los cuales se le

ha dado acceso a diseñadores a bases de datos, y para continuar explorando
el tema, hemos decidido crear una segunda versión del primitivo sistema que
tenemos ahora. Una de las cosas que nos ha hecho mucha falta es un sistema
de búsqueda 'inteligente' que le dé soporte al usuario en sus consultas, ampliándolas o reduciéndolas según sea necesario, mostrándolas en un formato más amable y, eventualmente, sugiriéndole de manera proactiva otros
términos de búsqueda u otra información que no haya sido requerida pero que
pueda ser relevante.

Lo que esto dice es que, en principio, el sistema va a ser usado en "condiciones de laboratorio" para experimentos. Yo espero, sin embargo, que
en el futuro pueda tener aplicaciones reales en, digamos, estudios de diseño. En una situación típica experimental, no creo que hayan más de 10 usuarios concurrentes. En una sesión de trabajo de 4 horas con 3 usuarios concurrentes hemos tenido unos 300 accesos al sistema (navegación+búsqueda)

>3. Me puedes mandar las especificaciones de la maquina sobre la cual
>estan montando el sistema.

Pentium IV 1.5GHz, 512MB Ram, 80GB HD
Linux Red Hat 8.0
Apache 2.0
MySQL Ver 11.18 Distrib 3.23.54
PHP 4.2.2
Server Name: dutodj.io.tudelft.nl

>4. El sistema de informacion que ustedes tienen tiene un nombre
>definido?

Nop, estamos buscando uno. Alguna idea?

>5. Tienes alguna propuesta para el nombre del proyecto?

Te refieres a tu proyecto de grado? Hmmmm, no. Déjame pensar en eso y te cuento

>Esto es todo por el momento. Gracias por tu ayuda.
>
>Mauricio

Chao,

_____
John Restrepo <j.restrepo@io.tudelft.nl>
TU Delft
Industrieel Ontwerpen
Landbergstraat 15 Kamer 10-4B-39
2628CE Delft
The Netherlands
Phone +31 15 278-5146 Fax +31 15 278-7179
_____

10 BIBLIOGRAFÍA

[1] APACHE FUNDATION. Servidor Web Apache [on line]. http://www.apache.org (Consulta Junio 2003)

[2] BAEZA YATES, Ricardo. RIBEIRO NETO, Berthier; Modern Information Retrieval; Editorial Addison Wesley.

[3] BETANCUR TORO, Diana Cristina. Web Semántica. Universidad EAFIT

[4] CONTENT MANAGER (UNIÓN EUROPEA). A European resource for Content Managers and CMS Suppliers [on line], http://www.contentmanager.eu.com (Consulta Julio de 2004)

[5] COPELAND, Jack. What is Artificial Intelligence?. [on line] http://www.alanturing.net/turing_archive/pages/Reference%20Articles/What%20is%20AI.html

[6] Ez Systems. Gestor de Contenidos Ez Publish [on line]. http://www.ez.no (Consulta Marzo de 2003)

[7] FOLEY, Jim. Computing vs. Computer science. College of computing [on line]. http://www.cra.org/reports/computing/ (Consulta Mayo 2004)

[8] IHMC Institute for Human and Machine Cognition [on line]. http://www.ihmc.us (Consulta Agosto 2003)

[9] MCCARTHY, John. What is Artificial Intelligence?. Universidad de Stanford [on line]. http://www.kurzweilai.net/articles/art0088.html?printable=1 (Consulta Junio 2004)

[10] MCDONALD, Mike. Librerías Socketcc y Pthreadcc. Monash University Australia [on line]. http://www.monash.edu.au (Consulta Junio 2004)

[11] MySQL. Base de Datos MySQL [on line], http://www.mysql.com (Consulta Julio 2003)

[12] PHP. Proyecto PHP [on line]. http://www.php.net (Consulta Mayo 2003)

[13]POSTGRE. Base de datos PostgreSQL [on line]. http://www.postgre.org (Consulta Junio 2004)

[14] ROBERTSON, James. So, What is a content management system? [on line]. http://www.steptwo.com.au/papers/kmc_what/

[15] TIPO 3 CMS. What is a Content Management System?. Tipo 3 Org [on line]. http://typo3.com/What_is_a_CMS_.1351.0.html (Consulta Marzo 2004)

[16] WITTEN, Ian H. Managing Gigabytes. Morgan Kaufmann Publishers.

[17] WORDNET PROJECT. Universidad de Princeton [on line]. http://cogsci.princeton.edu/~wn/ (Consulta Marzo de 2003)

[18] ZAMORA, Sergio. ¿Qué es la semántica?. Sitio público GEOCITIES [on line]. http://www.geocities.com/sergiozamorab/semantic.htm (Consulta Junio 2004)