UNIVERSIDAD POLITÉCNICA DE PUEBLA

PROGRAMA ACADÉMICO DE
INGENIERÍA EN INFORMÁTICA

Diseño de un Método de Búsqueda de Información Científica en la Web

José Manuel Martínez Rojas

Reporte Técnico PII-11-04-09

COMITÉ EVALUADOR

Dra. Rita Marina Aceves Pérez (*Asesor*)
M.C. Rebeca Rodríguez Huesca (*Sinodal*)
Dra. María Auxilio Medina Nieto (*Sinodal*)

PROFESOR(A) DE PROYECTO DE INVESTIGACIÓN II

Dra. María Auxilio Medina Nieto

Juan C. Bonilla, Puebla
Abril 2009

Índice

Resumen

La web es un repositorio que crece continuamente, hacer búsquedas de artículos científicos en él cada vez es más difícil, ya que hay muchos anuncios y documentos con información no relevante. Para afrontar este problema, este proyecto describe el diseño y la implementación de un metabuscador de carácter científico que procesa los resultados de los buscadores Google Académico, Scirus y Citeseer.

Capítulo 1. Planteamiento del Problema de Investigación

1.1 Introducción

El tesoro más valioso de la humanidad es el conocimiento y la investigación científica es, por excelencia, el mecanismo encargado de su generación [Guzmán R. et. al, 2005]. Una de las etapas principales de toda investigación científica es la revisión de los conocimientos previos. Tradicionalmente, esta revisión se hace a partir de publicaciones impresas tales como libros y revistas; sin embargo, debido al crecimiento de la web, las publicaciones en línea son cada día más utilizadas como punto de referencia.

Las máquinas de búsqueda disponibles son suficientemente generales para auxiliar en todo tipo de consultas, pero son inapropiadas para buscar información específica como aquella de tipo científico. Este proyecto pretende diseñar e implementar un metabuscador especializado en descubrir información científica disponible en la web, además de que permitirá obtener mejores resultados en comparación con sistemas motores de búsqueda tradicionales como Google o Yahoo. Actualmente existen varias máquinas de búsqueda de información científica en la web como CiteSeer[1], Sirus[2], o Google Académico[3], pero en su mayoría limitan la búsqueda a dominios específicos y/o colecciones en línea. El método propuesto empleará buscadores, metabuscadores y técnicas recuperación de información.

[1] http://citeseer.nj.nec.com/
[2] http://www.scirus.com/
[3] http://scholar.google.es/

1.2 Objetivo General

- Diseñar un método de *búsqueda información científica en la web* que permita identificar publicaciones científicas relevantes a partir de la salida de varios buscadores y metabuscadores científicos.

1.3 Objetivos Específicos

- Diseñar un método para la fusión de los resultados de diferentes buscadores de información científica.

- Implementar un prototipo para la recuperación de información científica que haga uso de varios buscadores científicos.

1.4 Justificación

La web se ha convertido en los últimos años en la principal fuente de información para cualquier tipo búsqueda, en donde el crecimiento explosivo en cuanto al número de datos hace difícil realizar una búsqueda que devuelva información útil, por lo que se ha hecho necesario el desarrollo de tecnologías que permitan localizar información científica de forma rápida y cómoda. Los motores de búsqueda tradicionales como Google, Yahoo, entre otros, no siempre proporcionan resultados relevantes, ya que muchos de los documentos que devuelven no son de calidad o no tiene relación con la búsqueda solicitada por el usuario.

El método que se propondrá hará la recuperación a partir del procesamiento de los resultados de varios motores de búsqueda y metabuscadores, ambos tipos de carácter científico. Se proporcionará una consulta y se funcionarán los resultados que se recuperen de cada uno de los buscadores, ahorrando así tiempo, ya que el usuario no tendrá que realizar varias búsquedas por separado para obtener información relevante.

1.5 Cronograma

La Tabla 1 muestra el cronograma con las actividades principales para llevar a cabo este proyecto.

Tabla 1. Cronograma de actividades

		Septiembre				***Octubre***				***Noviembre***				***Diciembre***		***Enero***			***Febrero***				***Marzo***				***Abril***	
No.	***ACTIVIDAD***	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*	*26*	*27*
1	*Elaboración de propuesta de investigación.*																											
2	*Revisión de literatura*																											
3	*Selección de motores de búsqueda y metabuscadores*																											
4	*Análisis de funcionamiento de metabuscadores y buscadores*																											
5	Presentación de propuesta de investigación.																											
6	*Análisis de técnicas de recuperación de resultados*																											
7	*Pruebas de técnicas de fusión*																											
8	Implementación de recuperación de información de buscadores científicos																											
9	Presentación de protocolo de investigación																											
10	Análisis de la aplicación																											
11	Diseño de la aplicación																											
12	Pruebas de la aplicación en forma local																											
13	Pruebas de la aplicación en forma remota (servidor web)																											
14	Presentación de proyecto de Investigación																											

1.6 Recursos

Hardware:

- Disco duro de 10 GB
- Memoria RAM: 256 MB
- Procesador con una velocidad mínima de 1 GHz

Software:

- Navegador de Internet Explorer versión 5 en adelante ó Firefox
- Servidor de Apache Tomcat para aplicaciones JSP versión 6
- JDK versión 1.6
- Sistema Operativo Windows XP

1.7 Alcances y limitaciones

Alcances

- Se realizarán búsquedas de contenido científico.
- El método aceptará cualquier idioma para realizar las búsquedas.
- El método tendrá crecimiento a futuro para hacer extracción de datos y obtener mejores resultados.

Limitaciones

- Por el momento, sólo se trabajará con la recuperación de la información de los buscadores y metabuscadores científicos siguientes: CiteSeer, Sirus, y Google académico.
- Sólo se tendrá acceso a publicaciones gratuitas.

Capítulo2. Marco Teórico

2.1 Introducción a la búsqueda de información en la web

Las herramientas de búsqueda de información más comunes son los motores de búsqueda, también llamados buscadores, los cuales están instalados en un servidor remoto. Ofrecen diferentes tipos de búsqueda, desde el formato del archivo, hasta el idioma.

Con el enorme crecimiento que la web ha experimentado, el problema que plantean los servicios de información creados y mantenidos "a mano" es que apenas pueden abarcar una parte mínima de los contenidos. Los motores de búsqueda cubren de un 20-25% de la web, mientras que los principales índices es dudoso que lleguen a un 5% [Lara N., Martínez U., 2006]. La desventaja de este extenso volumen de información que hay en la web es que, por razones de velocidad en la respuesta, obliga a limitar los servicios de búsqueda.

2.1.1 Búsqueda de información científica en la web.

La necesidad de buscar información científica en la web hace que se desarrollen nuevas tecnologías y que se adopten nuevos métodos de búsqueda, los cuales ayuden al descubrimiento de contenido importante que se encuentra oculto y por tanto, que no se ocupa. En la actualidad, las nuevas herramientas utilizadas en la máquina cliente hacen que sea más fácil el manejo de grandes cantidades de información, automatizando tareas que incrementan la productividad final de los recursos recuperados [Codina L., 2007].

Además del volumen de los datos en la web, hay que sumar la dificultad para obtener resultados académicos o científicos cuando se utilizan términos que se escriben igual pero tienen diferente significado (palabras homógrafas) que otros términos propios del comercio o de la cultura popular. Por ejemplo, a alguien muy interesado en la fisiología del sueño le resultará muy difícil encontrar información

sobre la fase del sueño denominada Rapid Eye Movement y que se conoce internacionalmente como REM, ya que si introduce dicha expresión en Google solamente, encontrará resultados relacionados con el grupo musical. La palabra clave "Dolly" proporciona otro buen ejemplo: si alguien está interesado en clonación y quiere informarse sobre el famoso experimento de clonación de la oveja Dolly, es probable que en un motor de búsqueda como Google solamente encuentre información sobre la cantante Dolly Parton.

2.2 Herramientas de recuperación de información

En la actualidad, existe una gran cantidad de herramientas de recuperación de información en la web como los directorios o índices temáticos, agentes buscadores, metabuscadores, motores de búsqueda, entre otras. Las secciones siguientes describen algunas de las más comunes, los motores de búsqueda y los metabuscadores.

2.2.1 Los motores de búsqueda

Los buscadores son bases de datos creadas por indización automática del texto completo de las páginas web y realizadas por un programa llamado robot. Este robot lógico, araña o spider, explora de forma automática los servidores, extrayendo las palabras más significativas de cada página y creando un índice de búsqueda. Según [Merlino S.,2001], la terminología que se usa en inglés para denominar a los robots (Spiders, wanderers, worms o crawlers) hacen que parezca que éstos viajan en a través de toda la World Wide Web, lo cual es un pensamiento erróneo, ya que los robots son estáticos, no tienen la capacidad de moverse de una computadora a otra pero pueden hacer uso de los recursos de la red para acceder a recursos remotos. Aún cuando los robots se comportan de forma similar, no existen dos robots de búsqueda exactamente iguales en términos de tamaño, velocidad y contenido; no existen dos motores de búsqueda que utilicen coincidentemente el mismo listado de relevancia y tampoco cada motor de búsqueda ofrece sus propias opciones de

búsqueda. Por lo tanto, la búsqueda depende del motor utilizado. La diferencia podría no ser mucha, pero sí significativa.
Existe una gran porción de la red en la que los “robots” de los buscadores no pueden o no alcanzan a indizar, a estos se les nombra como la "Red Invisible " o la "Red profunda" e incluye, entre otras cosas, sitios protegidos por contraseñas, documentos detrás de “cortinas de fuego”, material archivado, herramientas interactivas y contenidos de ciertas bases de datos [Lara N., Pablo M., 2004].

2.2.1.1 Funcionamiento de los motores de búsqueda

Un motor de búsqueda está formado por cuatro elementos básicos [Lara N., et. al, 2006]:

1. Un *robot* que recorre la red buscando recursos de información y sus respectivas URLs.

2. Un *sistema automático de análisis de contenidos e indexación* de los documentos localizados por el robot.

3. Un *sistema de interrogación*, generalmente basado en la lógica booleana, que permite al usuario expresar su consulta.

4. Un programa que actúa como intermediario entre el servidor de documentos HTML y la base de datos.

El funcionamiento de un motor de búsqueda es como sigue: el motor recibe la consulta del usuario (query) formada por uno o más términos, realiza una consulta interna en la base de datos que contiene los recursos web indexados y ofrece una lista de aquellos recursos que cumplen una parte o el total de los requisitos establecidos en la consulta. Generalmente, los resultados aparecen ordenados según una puntuación (score) que el programa asocia automáticamente a cada recurso, (ver la Figura 1).

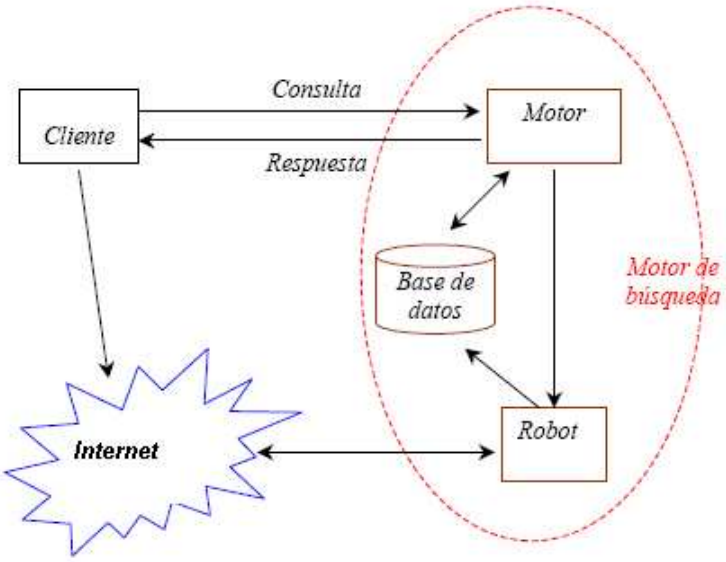


Figura 1. Estructura general de un motor de búsqueda [Merlino S. 2001].

Para realizar una búsqueda es necesario tener en cuenta un conjunto de variables:

1. Lenguaje que debe ofrecer diferentes tipos de operadores: lógicos, de comparación, de truncamiento, de proximidad o de especificación de campo.

2. Posibilidad de refinar una búsqueda inicial.

3. Campos limitadores que permitan reducir la búsqueda: dominios, lenguas, países, fecha de creación del recurso.

4. Búsquedas alternativas: búsqueda simple, búsqueda avanzada, búsquedas combinando operadores e índices temáticos.

5. Opciones avanzadas como: 1) buscar diferentes recursos (texto, sonido, imagen), 2) guardar búsquedas, 3) utilizar diferentes formatos en los resultados de búsqueda (estándar, detallado, compacto), 4) búsqueda de conceptos relacionados (related topics), 5) consulta directa en bases de datos (intranets), entre otras.

2.2.2 Metabuscadores

Un metabuscador es una herramienta de búsqueda que envía la petición simultáneamente a varios motores de búsqueda y así automatiza el proceso de realizar una misma consulta en diversos motores, lo cual no significa que sea totalmente exhaustivo. En las búsquedas se incluyen algunas veces lo que es llamado webs invisibles. ¿????EXPLICAR AQUÌ A QUÉ SE LE LLAMA UN WEB INVISIBLE..,

Después de haber recogido los resultados, los metabuscadores eliminan los enlaces duplicados y de acuerdo con los algoritmos implementados, hace una combinación; como se muestra en la Figura 2.

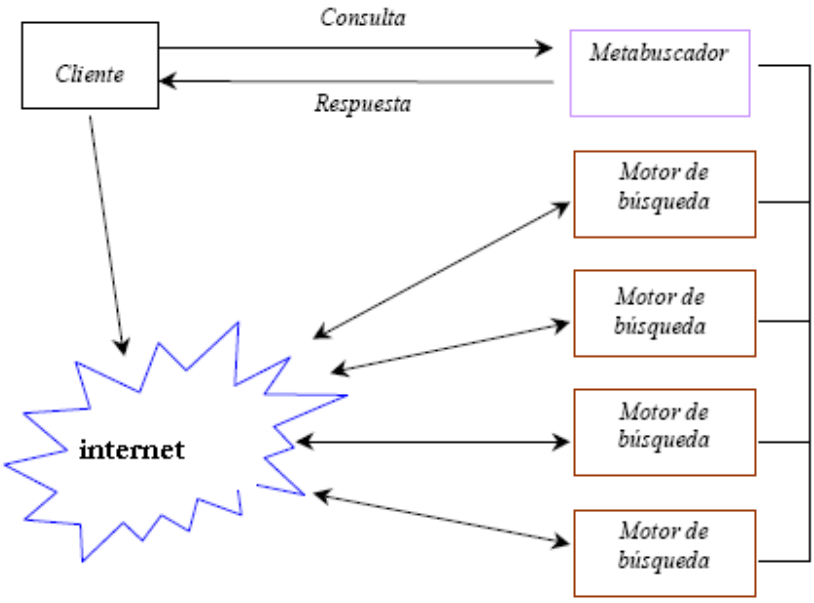


Figura 2. Estructura general de un motor de búsqueda [Merlino S., 2001].

2.2.2.1 Funcionamiento de los metabuscadores

En los siguientes pasos se describe el principio general del funcionamiento de los metabuscadores [Merlino, 2001]:

a) Aceptar la búsqueda que desea el usuario. mEl usuario introduce la palabra o frase a buscar en el formulario de búsqueda y se realiza el envío tecleando aceptar.
b) Convierte la palabra o frase a buscar a la sintaxis correcta para cada motor de búsqueda.
c) Remite la consulta en sus múltiples sintaxis a los diversos motores.
d) Espera por las respuestas un tiempo prudente para recoger la totalidad de las respuestas brindadas por las herramientas de recuperación.
e) Captura los resultados y elimina los URL duplicados.
f) Combina los resultados en donde todos los aciertos son mezclados conformando una única lista de ítems.
g) Elabora el ranking y ordena la lista por relevancia de los documentos.
h) Entrega los resultados postprocesados al usuario y presenta la lista final al usuario como respuesta a su búsqueda.

2.2.2.2 Tipos de metabuscadores

En la actualidad existen dos tipos de metabuscadores [Merlino S.,2001]:

- Lado del servidor.- Se ejecutan desde un servidor de manera remota
- Lado del cliente.- Se instalan y se ejecutan desde la computadora cliente donde están instalados sin necesidad de conectarse a ningún sitio alojado en la web.

2.3 Fusión de respuestas

Para obtener una única lista de documentos, es necesario fusionar las listas individuales de resultados que proporcionan los buscadores o metabuscadores científicos. A continuación se mencionan algunos de los algoritmos usados para la fusión de documentos.

2.3.1 Algoritmo básico CVV

El algoritmo CVV, también llamado “el método que alineaba CVV”, fue desarrollado por Budi Yuwono (de la Universidad de Ohio) y Dik Lee (de la Universidad de Hong Kong de la ciencia y de la tecnología). CVV fue diseñado para el uso con motores de búsqueda de Internet. Dado una pregunta, el algoritmo básico CVV calcula los méritos (también llamados las cuentas de la “calidad”) para cada colección en el sistema. Estos méritos estimados consisten en sumas de los productos, cada producto que contenga “los componentes”: un componente CVV y un componente del DF como se muestra a continuación: ¿????HAY UN CUADRITO EN LA FÓRMULA, DESPUÉS DEL SIGNO =, ES CORRECTO?

Calculo básico de mérito
$merito_{query,coll} = \sum_{term\ in\ query} (CVV_{term} * DF_{term,coll})$

donde $DF_{term,\ coll}$ es la frecuencia del documento, el número de documentos en la colección en la cual el término ocurre por lo menos una vez. N_{coll} es el número total de documentos en la colección, y por lo tanto de servicios, es límite superior para DF. El límite inferior para lDF es 0.

CVV_{term} es la variación de la señal-validez, y representa la cantidad total de variación en densidades [DF-basadas] del término entre las colecciones. $IntD_{term,\ coll}$ es la

densidad interna, la proporción de los documentos en la colección que contienen un término. Asimismo, $ExtD_{term,\ coll}$ es la densidad externa, la proporción de los d*ocumentos no en la colección* ¿????CORREGIR REDACCIÓN pero que contienen al término. $CV_{term,\ coll}$ es la señal-validez, que expresa la densidad relativa. C es el sistema de colecciones en el sistema, para las descomposiciones de SYM y de UDC, |C| es 236. $CVV_{term,\ coll}$ es la variación en $CV_{term,\ coll}$ y se calcula así:

Componente de cálculo CVV_{term}
$\mathbf{IntD}_{term,coll} = DF_{term,coll} / N_{coll}$
$\mathbf{ExtD}_{term,coll} = \sum_{c \neq coll} (DF_{term,c}) / \sum_{c \neq coll} (N_c)$
$\mathbf{CV}_{term,coll} = IntD_{term,coll} / (IntD_{term,coll} + ExtD_{term,coll})$
$\mathbf{avgCV}_{term} = \sum_{coll\ in\ C} (CV_{term,coll}) / \mathbf{\lvert C \rvert}$
$\mathbf{CVV}_{term} = \sum_{coll\ in\ C} (CV_{term,coll} - avgCV_{term})^2 / \mathbf{\lvert C \rvert}$

2.3.2 CombSUM

CombSUM (CSUM) - *Tras Lee ¿????QUÈ ES ESTO?*, que asigna una puntuación de 31 - i para clasificación de documentos i' de los 30 primeros de cada motor, es decir, el principio el documento anotó 30, en el segundo anotó 29, y así sucesivamente. Todo documento no clasificado en el principio *30 se anotó 0.* ¿???? NO SE ENTIENDE QUÈ ES LO QUE ESTÁ EN CURSIVAS. A continuación, añade los resultados junto a los tres motores. Por ejemplo, si un documento es 10'th clasificado por yahoo, 25'th por buscar???? AQUÌ DEBE IR EL NOMBRE DE UNA MÁQUINA DE BÚSQUEDA, NO LA PALABRA BUSCAR, y 40'th por google, entonces su puntaje combinado es (31 - 10) + (31 - 25) + 0 = 21 + 6 + 0 = 27.

2.3.3 CombMNZ

Se asignará un total de la puntuación como en CombSUM *y, a continuación, multiplicado por el número de puntuaciones distinto de cero.* ¿???? NO SE ENTIENDE!..,En el ejemplo anterior, el documento es distinto de cero a dos puntuaciones, por lo que su puntaje combinado es de 2 (21 + 6 + 0) = 54.

2.3.4 Algoritmo 2-step RSV

El cálculo de la relevancia documental en dos pasos o 2-step RSV consiste en agrupar las frecuencias documentales de un término dado una consulta [Martínez-Santiago et al., 2003]. El método requiere calcular la puntuación obtenida por cada documento cambiando la frecuencia documental de cada término que aparece en la consulta: dado un término de la consulta, su nueva frecuencia documental seria el resultado de sumar a su frecuencia documental original, la frecuencia documental alcanzada por tal término en el resto de las sub colecciones seleccionadas. Por ejemplo, si las colecciones I_1 e I_2 son seleccionadas, y la consulta contiene el termino "gobierno", entonces la nueva frecuencia global (df) será df_{I_1}(gobierno) + df_{I_2}(gobierno). Dada una consulta los dos pasos son:

1. La fase de preselección de documentos corresponde con el lanzamiento de la consulta sobre cada subcolección seleccionada I_j. Como resultado de unir los documentos más relevantes recuperados, se obtiene para cada colección una única lista de documentos preseleccionados I^1.

2. La fase de reordenamiento consiste en volver a indexar la colección (ahora ya será un ranking) I^1, pero considerando tan sólo el vocabulario de la consulta. Finalmente, se elabora una nueva consulta formada por los términos anotados y se lanza tal consulta sobre el nuevo índice

2.4 Trabajo relacionado

Profusion es un metabuscador desarrollado en la Universidad de Kansas al cual se le envía una consulta, se analiza, se clasifica, y se eligen los motores de búsqueda para la consulta sobre la base de conocimientos a priori, que representa la idoneidad de cada motor de búsqueda para cada categoría (ver la Figura 4). Utiliza la fusión de los resultados para construir una nueva lista de peso de los documentos devueltos, elimina duplicados y, opcionalmente, los vínculos rotos. Presenta la clasificación final por orden de lista para el usuario (Figura 5). El rendimiento de Profusion se ha comparado con distintos motores de búsqueda y otros metabuscadores, demostrando su capacidad para recuperar información pertinente y presentar menos duplicación de páginas.

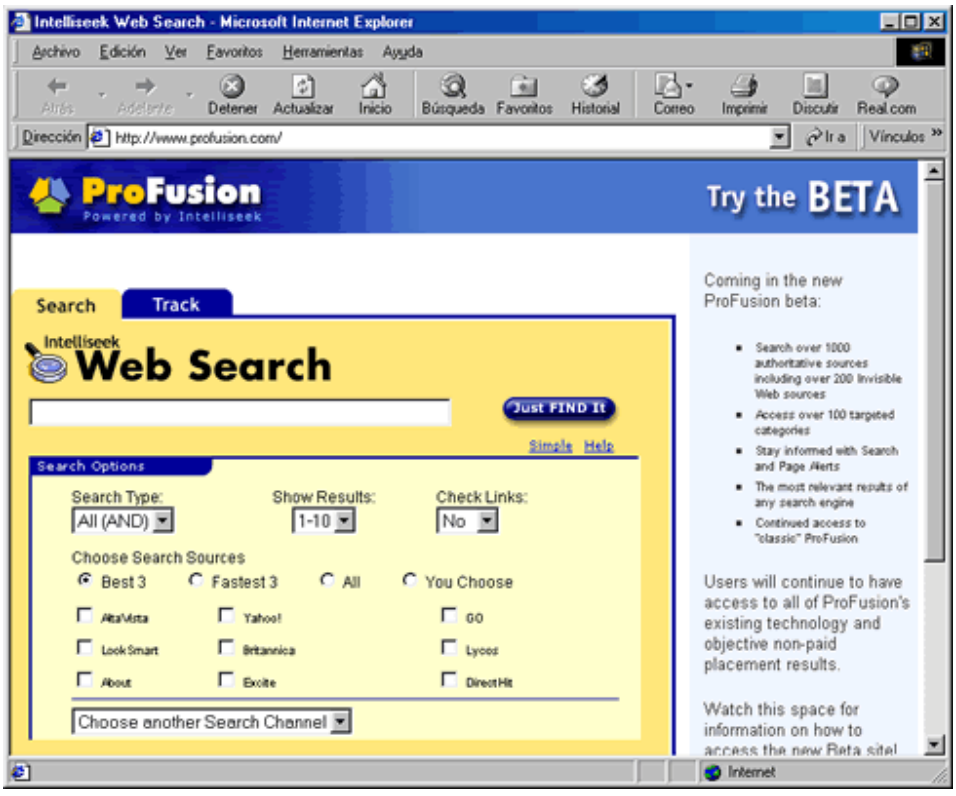


Figura 4. Opciones del metabuscador Profusion.

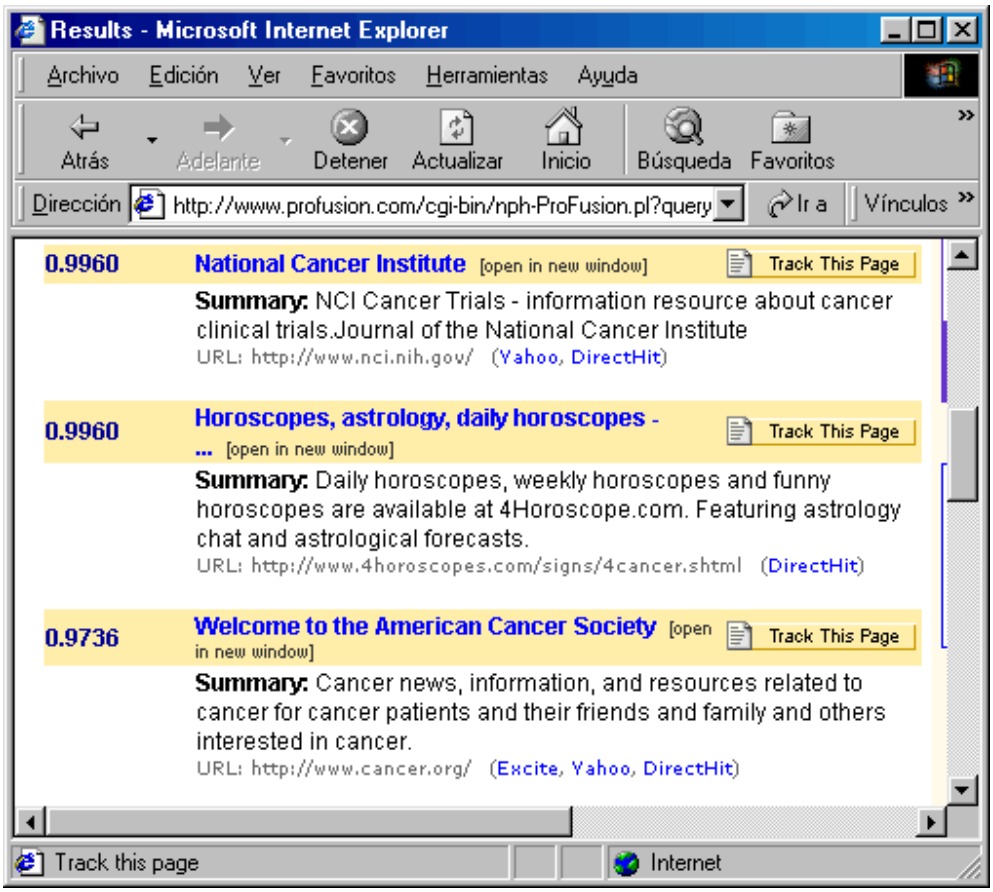


Figura 5. Resultado de una búsqueda en el metabuscador Profusion.

Las diferencias del metabuscador profusión y el metabuscador propuesto se muestran en la Tabla. En la actualidad existen metabuscadores que trabajan sobre buscadores comunes. El propuesto trabajará con motores de búsqueda y metabuscadores científicos,

Tabla 2. Diferencias entre el metabuscador Profusion y el metabuscador del proyecto

	Proyecto	**Profusion**
Lugar	UPP	Universidad de Kansas
Lenguaje	JSP	Perl
Plataforma de trabajo	Multiplataforma	Plataformas Unix
Buscadores	Scholar Google,Scirus, Citiceer	Google, Yahoo, altavista, OpenText, Infoseeck, exite…
Cantidad de buscadores	n	9
Buscadores predeterminados	Scholar Google,Scirus, Citiceer	Google, lycos, exite

Capítulo 3. Diseño

Este capítulo describe los elementos utilizados para el diseño del metabúscador de artículos científicos.

3.1 Requerimientos

Esta sección menciona cada uno de los requerimientos tanto funcionales y no funcionales que forman parte del sistema.

3.1.1 Requerimientos funcionales

El primer requerimiento funcional permite la realización de una consulta en el metabuscador. Su descripción está en la Tabla 3. Los requerimientos restantes están en las tablas 4-7.

Tabla 3. Requerimiento funcional realizar consulta

Descripción corta:	Representa la búsqueda que desea realizar el usuario
Descripción detallada:	1.- El usuario introduce la palabra o frase a buscar

Tabla 4. Requerimiento funcional recuperar la información de los buscadores y metabuscadores científicos

Descripción corta:	Representa la recuperación de las respuestas que obtiene cada buscador en base a la consulta realizada
Descripción detallada:	1.- Se recuperan los resultados de cada buscador 2.- Los resultados son almacenados en un archivo de texto.

Tabla 5. Requerimiento funcional eliminar respuestas repetidas

Descripción corta:	Representa la eliminación de respuestas repetidas
Descripción detallada:	1.- Se lee el contenido de cada archivo 2.- Si se encuentran respuestas repetidas en los archivos obtenidos de la recuperación, se mantiene sólo una

Tabla 6. Requerimiento funcional fusión de resultados

Descripción corta:	Representa la unión de los resultados almacenados en los archivos de texto.
Descripción detallada:	1.- Se lee el contenido de cada archivo 2.- Se unen las resultados de cada archivo de texto en nuevo archivo de texto

Tabla 7. Requerimiento funcional publicación de resultados

Descripción corta:	Muestra la unión de los resultados en una sola lista
Descripción detallada:	1.- Se lee el archivo de texto que tiene todos los resultados obtenidos 2.-Se muestra al usuario

3.1.2 Requerimientos no funcionales

Los requerimientos no funcionales que se han considerado para el diseño del metabuscador son los siguientes:

1. *Disponibilidad*: El metabuscador estará instalado en un servidor web, por lo que se accederá a él en línea

2. *Plataforma*: El metabuscador se desarrollará en JSP, por lo tanto, será multiplataforma debido a que JSP es una tecnología de Java.

3. *Calidad*. El sistema permitirá realizar búsquedas de documentos científicos y relevantes a la búsqueda solicitada.

4. *Escalabilidad*: Se planea que a futuro este proyecto sea capaz de adaptar nuevas funcionalidades.

3.2 Casos de uso

El comportamiento del metabuscador está documentado por los casos de uso de las tablas de esta sección.

Tabla 8. Caso de uso buscar artículos científicos

Caso de uso:	***CU_ Buscar artículos científicos***
Descripción:	El usuario introduce el tema o palabras clave a buscar
Actor principal:	Usuario
Personal involucrado e interés	Usuario: necesita obtener información científica.
Precondiciones:	Los buscadores estén activos.
Garantías de éxito: (post condiciones)	Los resultados que se obtengan serán confiables ya que provienen de buscadores que obtienen la información de fuentes científicas.
Escenario principal de éxito o flujo básico	1. Consulta de palabra o frase.
Extensiones o flujos alternativos	Problemas con el navegador a) Reintentar o probar otro navegador disponible b) Actualizar navegador Descarga incompleta del metabuscador ¿????PERO NO SE DESCARGA, O SI? ES DEL LADO DEL SERVIDOR. CORREGIR. a) Cerrar el navegador y volver a iniciar Problema con los datos a buscar

	a) No introducir algún dato.

Tabla 9. Caso de uso recibiendo lista de artículos

Descripción:	Los buscadores y metabuscadores devuelven al usuario respuestas de la búsqueda de las palabras clave.
Actor principal:	Usuario
Personal involucrado e interés	Usuario: Espera la lista de artículos científicos relacionados con las palabras clave de la consulta.
Precondiciones:	1.- Que el usuario haya escrito una palabra o frase 2.- Los buscadores que proporcionan las respuestas.
Garantías de éxito: (post condiciones)	El servidor que mantiene la aplicación estará activo cada vez que se requiera hacer una consulta.
Escenario principal de éxito o flujo básico	El usuario espera la respuesta de la consulta realizada.
Extensiones o flujos alternativos	Problemas con el navegador c) Reintentar o probar otro navegador disponible d) Actualizar navegador Descarga incompleta del metabuscador. b) Cerrar el navegador y volver a iniciar

3.3 Especificación de actores

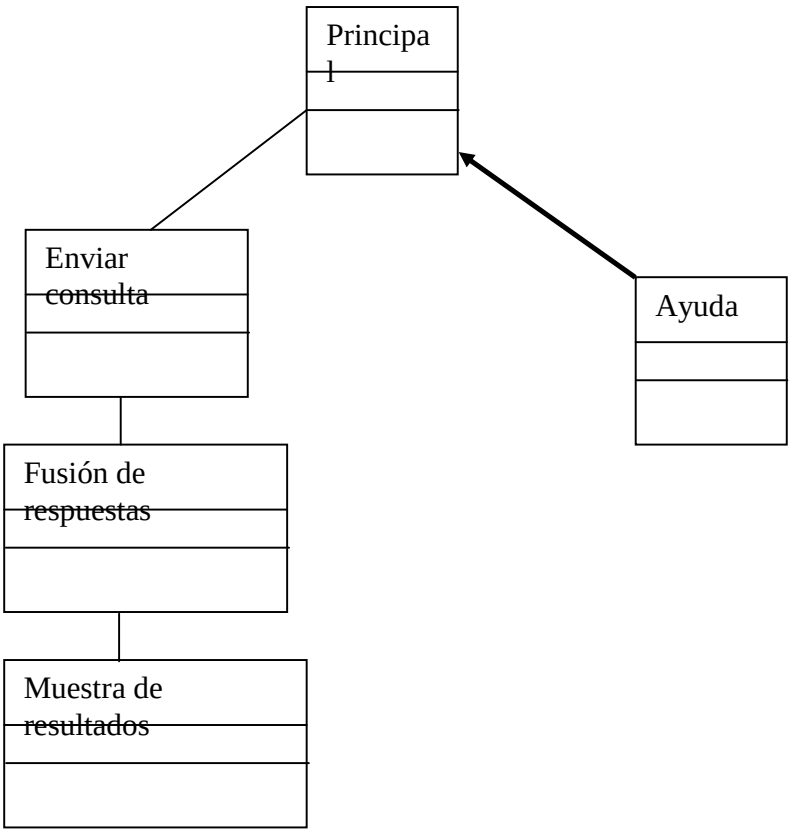
La Tabla 10 muestra la descripción del actor involucrado en el uso del sistema.

Tabla 10. Especificación del actor usuario

Descripción:	Persona que ingresa y consulta el metabuscador
Características:	Usuario que tiene la necesidad de buscar información científica
Relaciones:	Interactúa con el sistema para introducir y buscar la información

3.4 Diagrama conceptual

La Figura 6 muestra el diagrama conceptual del sistema, el cual indica que desde la página principal se mostrará un menú de ayuda o se iniciará una sesión.



¿???? CORREGIR LA FIGURA PORQUE HAY LÍNEAS EN LAS PALABRAS

Figura 6. Resultado de una búsqueda en el metabuscador Profusion.

3.5 Datos de entrada, salida y descripción por módulo

Las tablas 11- muestran los datos de entrada y salida de los módulos de la Figura 6.

Tabla 11. Descripción del módulo enviar consulta

Descripción	Contendrá la palabra a buscar y la enviará a los distintos buscadores
Datos de entrada	Palabra o frase.
Datos de salida	Archivo de respuestas por cada buscador

Tabla 12. Descripción del módulo fusión de respuestas

Descripción	Unirá los archivos que se generaron en el módulo "Enviar consulta".
Datos de entrada	Archivo de respuestas por cada buscador.
Datos de salida	Un sólo archivo con la unión de las respuestas de cada buscador.

Tabla 13. Descripción del módulo mostrar resultados

Descripción	Mostrará los resultados del archivo generado en el módulo de "Fusión de respuestas".
Datos de entrada	Archivo con la lista de resultados obtenidos.
Datos de salida	Se muestra el archivo de entrada a una página web.

3.6 Diseño de pantallas

Las figuras de esta sección muestran de manera gráfica el funcionamiento del metabuscador, la forma de realizar una consulta y la obtención de resultados.

3.6.1 Pantalla principal

La Figura 7 muestra la interfaz principal donde se realizan las búsquedas. Esta pantalla cuenta con lo siguiente:

- Enlace de **Acerca de**… que proporciona información acerca del metabuscador.
- Enlace de **Contacto** que muestra información para contactar al desarrollador del metabuscador.

- Título de la página. En la parte central se encuentra una animación hecha en flash en la cual se encuentran girando los nombres de los buscadores.
- La sección de búsqueda que contiene una caja de texto en la que se introduce la palabra o frase que se desea buscar.
- En la última parte se encuentra un menú en el cual se puede seleccionar entre uno o más buscadores para realizar la búsqueda y el botón de aceptar.

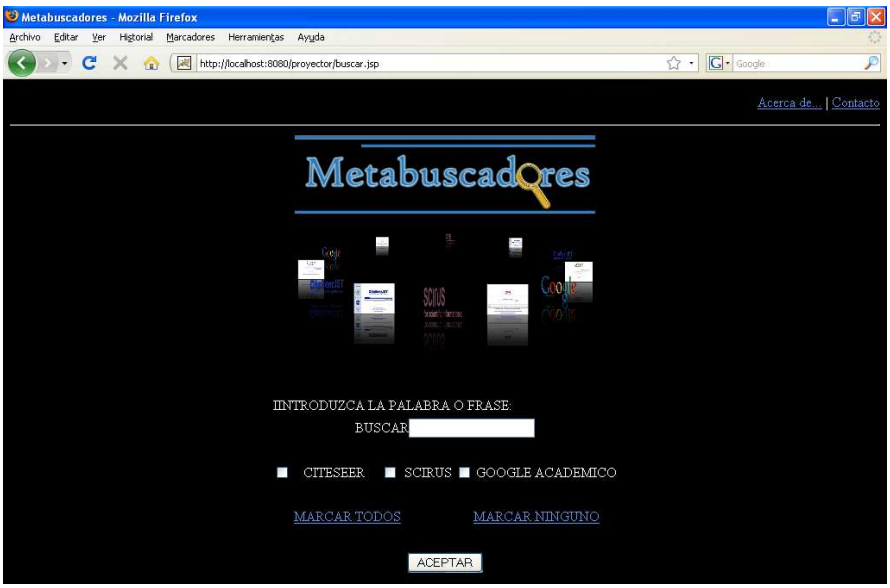


Figura 7. Pantalla principal

3.6.2 Pantallas de búsqueda y resultados

En esta sección se muestran pantallas que permiten observar el funcionamiento del metabuscador a través de ejemplos, además, muestran la forma de realizar una búsqueda así como.los resultados obtenidos. La Figura 8 muestra la pantalla principal con una consulta a realizar que es “digital image”, se tiene seleccionada únicamente la opción de citeSeer que es sobre la cual se realizará la búsqueda de información.

Figura 8. Pantalla de captura seleccionando citeSeer

La Figura 9 muestra la pantalla de los resultados obtenidos de la consulta realizada con “digital image” en citeSeer.

Figura 9. Pantalla de resultados

La Figura 10 muestra la selección de todos los buscadores que se realiza a través de la liga de *Marcar todos*.

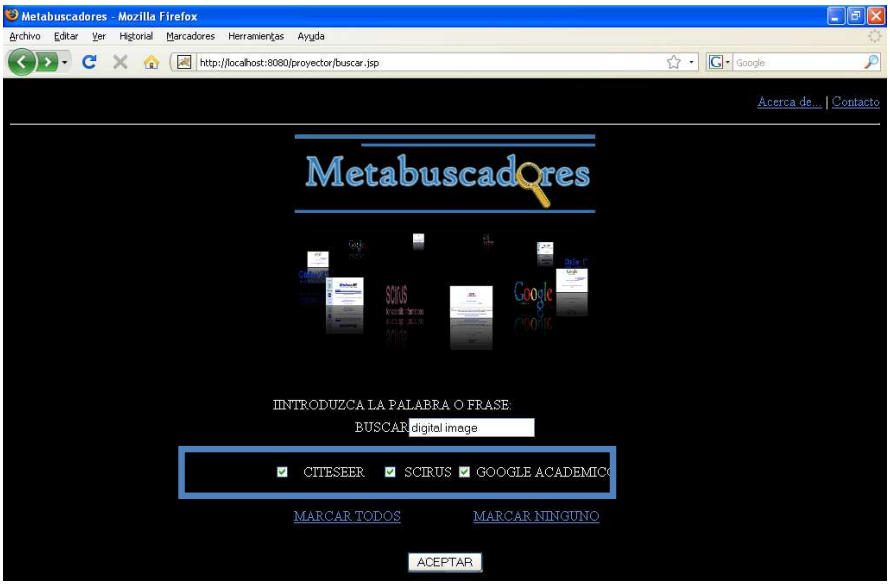


Figura 9. Consulta con los 3 buscadores

La Figura 10 muestra pantalla de los resultados con los 3 buscadores de forma oculta, con un enlace que al darle clic muestra los resultados por cada buscador.

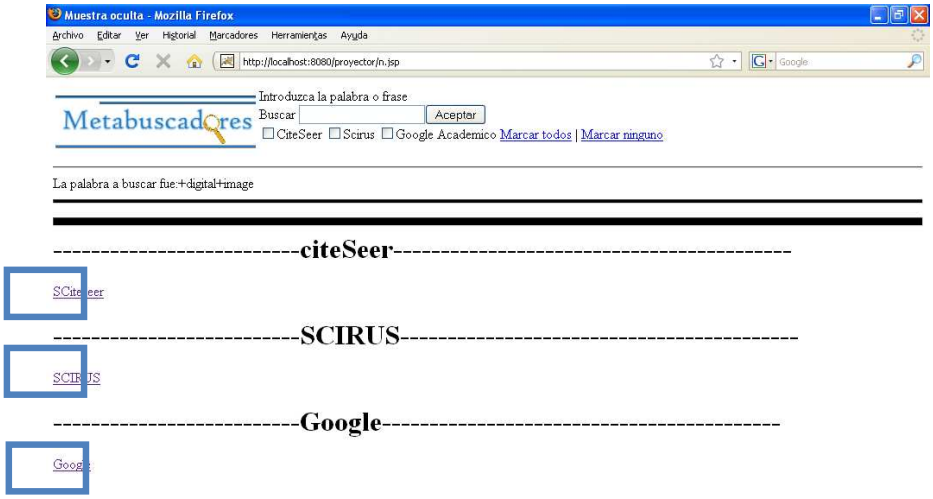


Figura 10. Resultados de los 3 buscadores ocultos

La Figura 11 muestra los resultados por cada buscador, en este ejemplo se han omitido algunos resultados, pero cada buscador muestra más de 3 resultados.

Figura 11. Pantalla de resultados con los 3 buscadores

3.6.3 Página de Acerca de…

La Figura 12 muestra la pantalla Acerca de… y su contenido. ¿????no tiene la versión del sistema, ni la del lenguaje y herramienta utilizadas

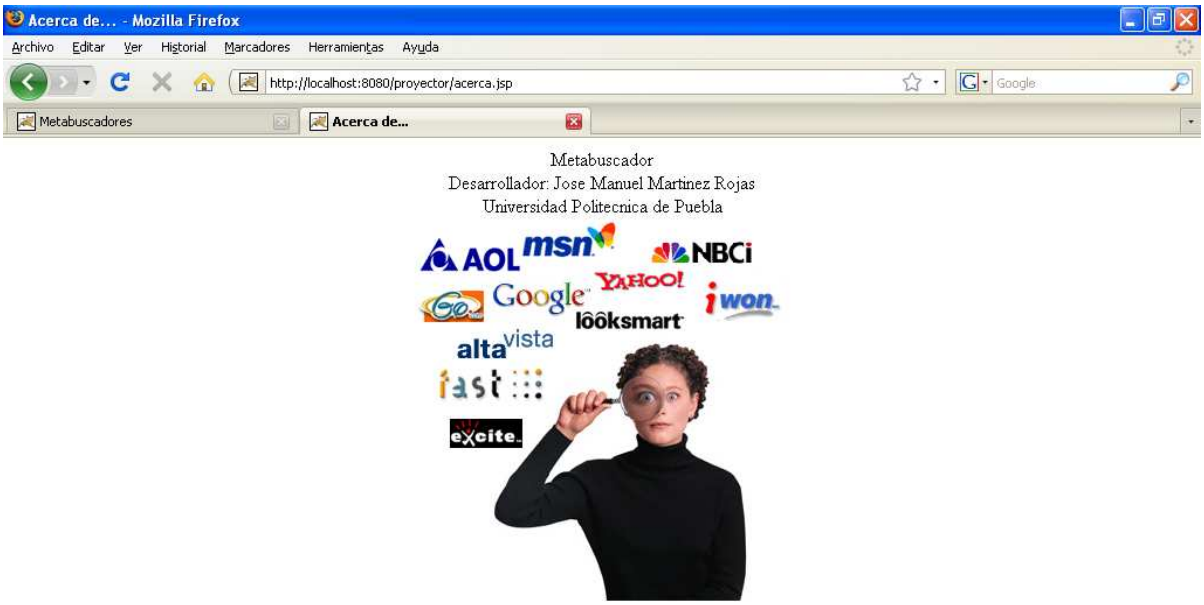


Figura 12. Pantalla Acerca de

3.6.4 Página de contacto

La Figura 13 muestra la pantalla de contacto y la información de ésta.

Figura 13. Pantalla de contacto

Capítulo 4. Implementación

En este capítulo se describe el funcionamiento e implementación del metabuscador desarrollado, el cual procesa la información a buscar sobre algún tema a través de frases o palabras clave.

4.1 Herramientas

Para el desarrollo e implementación de esta aplicación se utilizó el siguiente software:

a) JavaServer Pages (JSP)

   Es una tecnología Java que permite generar contenido dinámico para ambiente web, el cual permite mezclar HTML estático con HTML generado dinámicamente. Debido a la portabilidad que ofrece Java, se eligió esta tecnología, ya puede permite ejecutar la aplicación en cualquier plataforma mediante la máquina virtual.

b) Servidor Apache Tomcat 5.5

   Apache Tomcat es un servidor web con soporte para servlets y JSP's, el cual incluye un compilador de Java, convierte los archivos JSP en servlets. Se eligió esta tecnología porque la aplicación desarrollada se puede almacenar o montar en este servidor, se accede a ella desde cualquier lugar mediante un URL.

c) Python 2.5

   Python es un lenguaje de programación orientado a objetos, permite el uso de módulos, excepciones, sintaxis dinámica, clases y tipos de datos de alto nivel. Suele compararse con otros lenguajes, entre los que destacan Perl, Ruby o Java, es multiplataforma. Se hizo uso de este lenguaje de programación

debido a que el buscador Google Académico no permite la extracción de su página desde Java.

4.2 Fases desarrolladas

Las fases implementadas en el metabuscador son las siguientes:

1. Envió de información hacia los motores de búsqueda.
2. Extracción de información de los motores de búsqueda.

4.2.1 Envió de información hacia los motores de búsqueda

La primera fase es la extracción de los resultados de los motores de búsqueda, consiste en obtener los resultados generados por la consulta para su posterior procesamiento.

Para llevar acabo esta fase se observaron los patrones de acceso, que indican en qué parte se debe colocar la consulta, es decir, la palabra o frase a buscar. Las consultas o bases se muestran a continuación

a) `Citeseer` →`http://citeseer.ist.psu.edu/cis?q=`
b) `Scirus` →`http://www.scirus.com/srsapp/search?q=`
c) `Google escolar` →`http://scholar.google.com.mx/scholar?q=`

Una vez que se detectó la base de cada motor de búsqueda, se crearon los siguientes archivos JSP:

- Un archivo JSP llamado buscar.jsp el cual se encarga de recibir la cadena de búsqueda (una palabra o frase) y la envía a un segundo archivo JSP llamado n.jsp
- El archivo n.jsp procesa la cadena de búsqueda y extrae la información de los buscadores.

La cadena de búsqueda se procesa según el formato que utiliza la base de cada buscador. Primero se usa la función *split* para dividir las palabras de acuerdo al número de espacios que contenga la cadena como se muestra a continuación:

```
campos=cadena.split(" ");
```

donde cadena es la palabra que recibimos del formulario buscar.jsp y campos es una arreglo de tipo String, posteriormente se realizó la unión de todas las palabras separadas en el arreglo con el signo "+", el cual hace la función de espacio en la base de los motores de búsqueda. Esto se realizó con el ciclo siguiente:

```
for (int i=0;i<campos.length;i++)
    {
     cadena=cadena+"+"+campos[i];
    }
```

Por ejemplo: si tenemos la cadena "base de datos distribuidas" con la función split se tiene: cadena= base de datos distribuidas

Arreglo campos

Posición	Palabra
0	base
1	de
2	datos
3	distribuidas

Después de separar y concatenar cada una de las palabras de la cadena se tiene:

```
cadena= base+de+datos+distribuidas
```

Una vez que la cadena tiene el formato para la base, puede ser enviada hacia cada uno de los motores de búsqueda al concatenarla con una base, ejemplos:

Bases concatenadas:

a) Citeseer

→http://citeseer.ist.psu.edu/cis?q=base+de+datos+distribuidas

b) Scirus→http://www.scirus.com/srsapp/search?q=
base+de+datos+distribuidas
c) Google→http://scholar.google.com.mx/scholar?q=
base+de+datos+distribuidas

4.2.2 Extracción de información de los motores de búsqueda

El siguiente paso es hacer uso de la clase URL que permite la extracción del contenido de toda la página web del motor de búsqueda. Esta clase requiere la base concatenada y la cadena procesada. Ejemplo:

```
URL pagina =
new URL(http://citeseer.ist.psu.edu/cis?q= base+de+datos+distribuidas)
URL pagina = new
URL(http://www.scirus.com/srsapp/search?q=base+de+datos+distribuidas)
URL pagina = new URL(http://scholar.google.com.mx/scholar?q=
base+de+datos+distribuidas)
```

Después de hacer uso de la clase URL, se abre un objeto tipo BufferedReader para extraer el contenido de cada una de las variables “pagina” mediante la siguiente línea:

```
BufferedReader in= new BufferedReader(new
 InputStreamReader(pagina.openStream()));
```

.

Posteriormente, se lee el contenido de la variable in mediante un ciclo while y se almacena en la variable Stringbuffer mediante el método append para después poder crear un archivo de texto

```
StringBuffer buffer = new StringBuffer();
 while ((entra = in.readLine()) != null)
{
     buffer.append(entra);
}
in.close();
```

Hasta este punto sólo se han procesado las respuestas de los motores de búsqueda lo siguiente es guardar los resultados en un archivo de texto.

```
String str=new String(buffer);
BufferedWriter archi = new BufferedWriter(new
FileWriter("c:\\archivo2.txt",true));
archi.write(str);
archi.flush();
archi.close();
```

Primero se declara una variable llamada “archi” del tipo “BufferedWriter” con la función FileWriter que permite guardar en un archivo el contenido de la variable str, la cual contiene el valor de la variable buffer ya antes descrita, en este caso el archivo en el que se almacenará esta información es llamado archivo2.txt,. Con la función flush liberamos los recursos que hemos ocupado del sistema y con la función close cerramos el archivo.

La Figura 14 muestra el contenido del archivo2.txt, que es la extracción del contenido de la pagina de resultados de la consulta “base de datos distribuidas en citeSeer”:

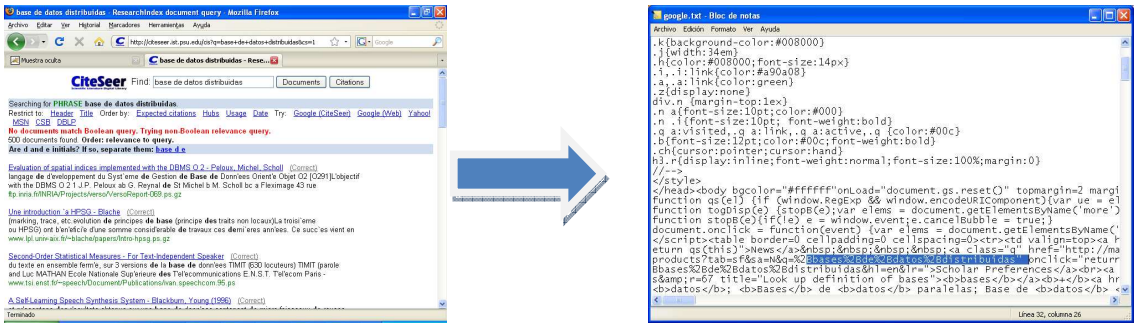


Figura 14. Contenido del archivo2.txt

El proceso de extracción mencionado es válido únicamente para los motores de búsqueda citeSeer y Scirus, ya que el acceso a Google es un poco más restringido, Para la extracción de la página de resultados de Google Académico se tuvo que hacer uso de otro lenguaje de programación que es pitón, este lenguaje hace uso de un parser llamado BeautifulSoup.py que contiene las instrucciones para extraer el

contenido de la página en código HTML. Para esto se crea desde el código principal de la extracción un archivo que contiene la cadena a buscar ya procesada, es decir con la sustitución de los espacios por el signo +, este archivo lleva por nombre python.txt

La Figura 15 muestra el contenido del archivo python.txt:

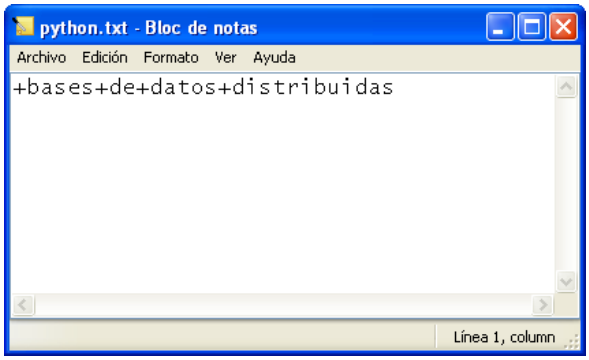


Figura 15. Contenido del archivo python..txt

Después se crea un archivo llamado recipe3.py el cual lee archivo python.txt que contiene la consulta y se encarga de extraer los resultados. El contenido del archivo python.txt es almacenado en una variable llamada “texto”.

```
archivo = open("c:\\python.txt", 'r')
texto = archivo.read()
```

Posteriormente se hace un llamado al BeautifulSoup con el parámetro de entrada que en este caso es la cadena a buscar mediante el siguiente código:

```
search = GoogleScholarSearch()
pubs = search.search([texto], 100)
```

donde:

- Primero se manda a llamar a la función encargada de extraer la información con BeautifulSoup.
- Después se le manda la palabra o frase a buscar acompañado de la cantidad de resultados a devolver, y por último se imprime la variable pubs.

Desde el programa principal escrito en JSP, se llama el archivo recipe3.py mediante la siguiente línea:

```
Process p = Runtime.getRuntime().exec ("cmd /c C:\\Python25\\recipe3.py >
                                        C:\\google.txt");
```

la cual crea un proceso llamado p para ejecutar el cmd de windows y hacer un redireccionamiento del resultado de récipe3.py a un archivo llamado google.txt para su posterior procesamiento.

La Figura 16 muestra el contenido del archivo google.txt:

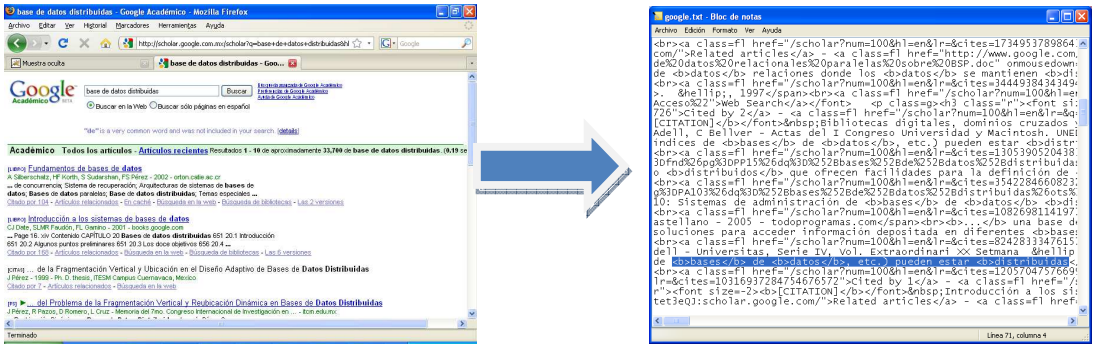


Figura 16. Contenido del archivo google.txt

Ahora se ha obtenido la página completa del motor de búsqueda que contiene los resultados, se procede a buscar patrones para extraer los resultados, éstos se muestran en la Tabla 14.

Tabla 14. Patrones de extracción

Buscador	Parámetro inicial	Parámetro final	Observaciones
Scirus	<table width=\"100%\" cellspacing=\"0\" cellpadding=\"0\" border=\"0\">	</table>	Delimita el área de resultados
Citeseer	<!--RLS-->	<!--RLE-->	Delimita el área de

			resultados
Google	<p class=g	</Font>	Delimita el área de resultados

Con los patrones de extracción se obtiene un área de trabajo limpia en la cual se trabaja únicamente con los resultados, los cuales están separados de acuerdo a alguna etiqueta HTML, por ejemplo, en el caso de los enlaces se emplea `<a href=`. Posteriormente cada uno de los resultados obtenidos es guardado en variables de tipo vector para su posterior procesamiento. Finalmente se imprimen los resultados obtenidos, cada resultado contiene lo siguiente:

- El título de la página que contiene el tema relacionado con la búsqueda.
- Un resumen acerca del tema.
- El enlace a la página original.

La Figura 17 muestra un ejemplo de la impresión de resultados obtenidos para Google Académico:

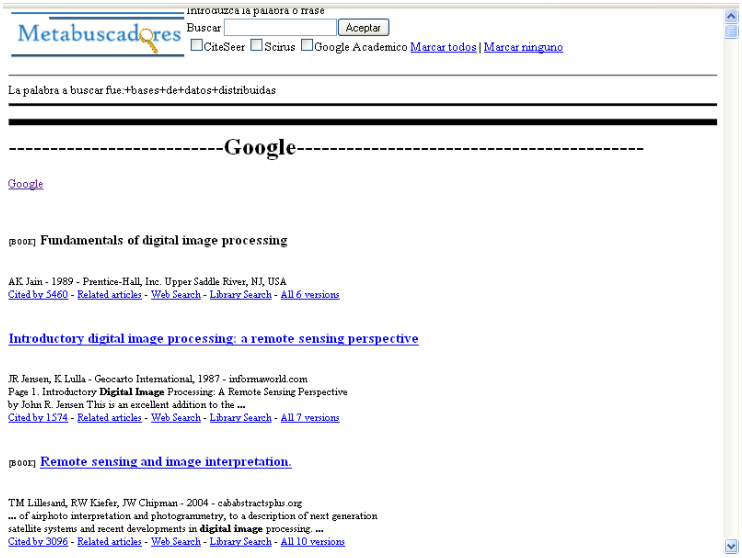


Figura 17. Resultados de Google Académico

Capítulo 5 Conclusiones

En este proyecto se desarrolló un metabuscador que emplea los resultados de los buscadores Google Académico, Scirus y Citeseer. Los buscadores proporcionan información de carácter científico, lo que hace relativamente confiable la información que se consulta en éstos. Para la construcción del metabuscador se identificaron cuatro etapas principales: 1) la extracción de los resultados de los buscadores, 2) la fusión de los resultados, 3) la eliminación de resultados duplicados, y por último 4) la impresión de los elementos más importantes de los resultados.

Entre los principales obstáculos que se encontraron para el desarrollo e implementación del metabuscador son que en ocasiones los buscadores no estaban disponibles, por ejemplo, la página principal del buscador Citeseer estuvo en mantenimiento durante un periodo, por lo que únicamente hacia búsquedas sobre el buscador Google. Otro problema importante fue que el buscador Google Académico está restringido y no permite la extracción de sus páginas con lenguaje Java. Sin embargo, utilizando el lenguaje Python si fue posible acceder a los resultados de este buscador, ya que sobre este lenguaje está desarrollado.

Desde la interfaz gráfica se puede seleccionar el buscador y realizar las búsquedas. Los resultados se muestran en la página siguiente de forma ordenada por buscador.

¿???????NO HAY TRABAJO A FUTURO??

¿??????? LAS REFERENCIAS VAN ANTES DE LOS ANEXOS

Anexo 1 Instalación de Apache Tomcat

En esta sección se muestra la forma de descargar e instalar Apache Tomcat, el cual es el servidor para aplicaciones Web en Java. Este programa se distribuye bajo licencia Apache Software License por lo que puede usarse en proyectos comerciales.
El proceso de instalación consta de las siguientes tareas:

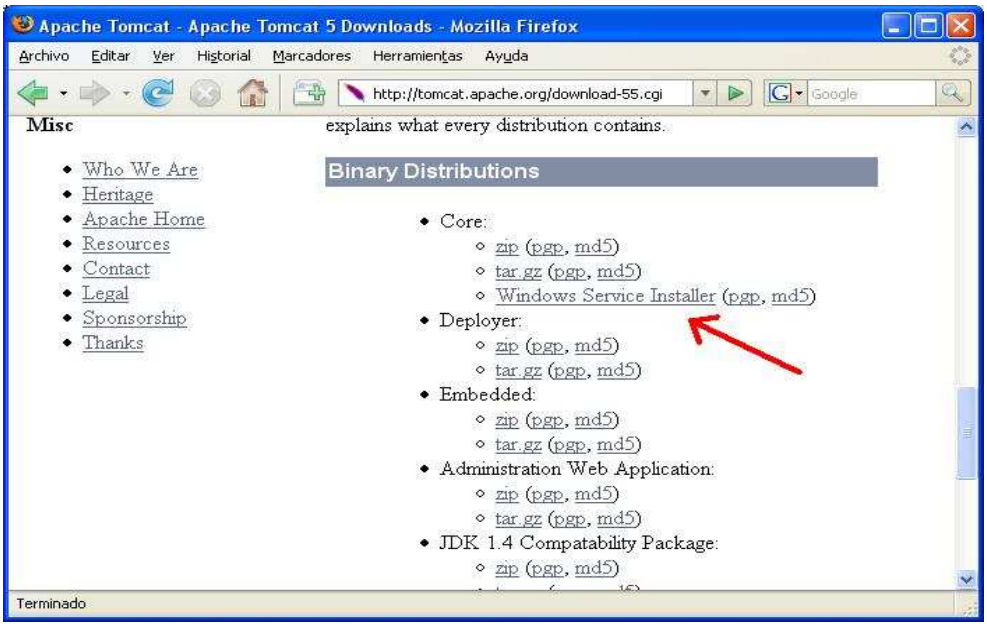
- Prerequisitos
- Descarga
- Instalación

Prerequisitos

Se necesita tener la Instalación del JDK

Descarga

Apache Tomcat se descarga desde la siguiente liga: http://tomcat.apache.org/download-55.cgi, de donde se Bajara hasta la sección “Binary Distributions” y descargara el fichero “Windows Service Installer”, en donde se descargara el fichero apache-tomcat-5.5.25.exe.

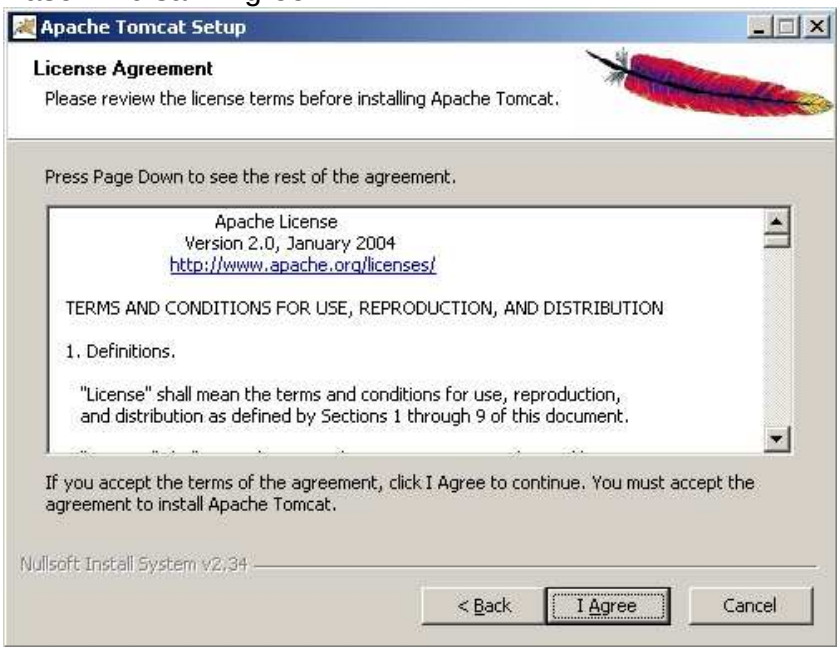


Instalación

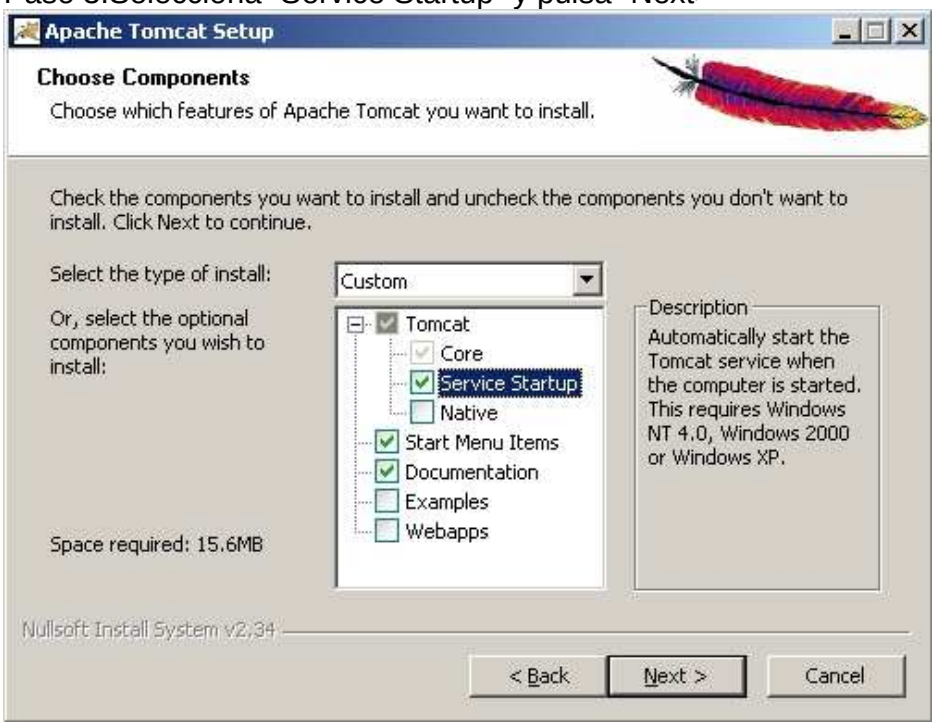
Se ejecutara el programa de instalación apache-tomcat-5.5.25.exe y se llevaran a cabo los siguientes pasos

Paso 1:Pulsa “Next>“

Paso 2:Pulsa “I Agree”

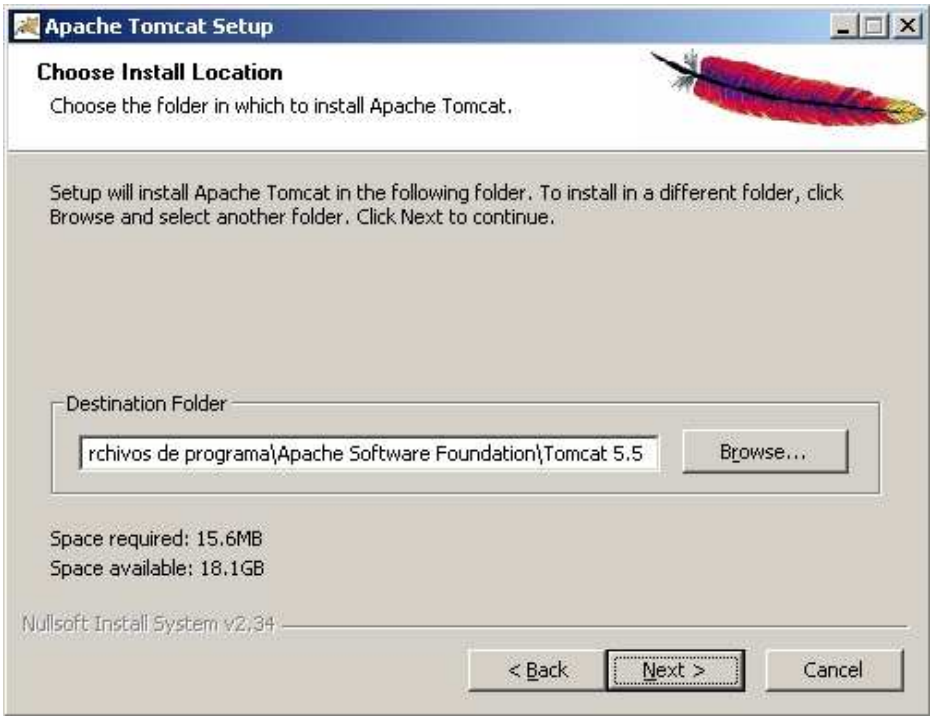


Paso 3:Selecciona “Service Startup” y pulsa “Next>“

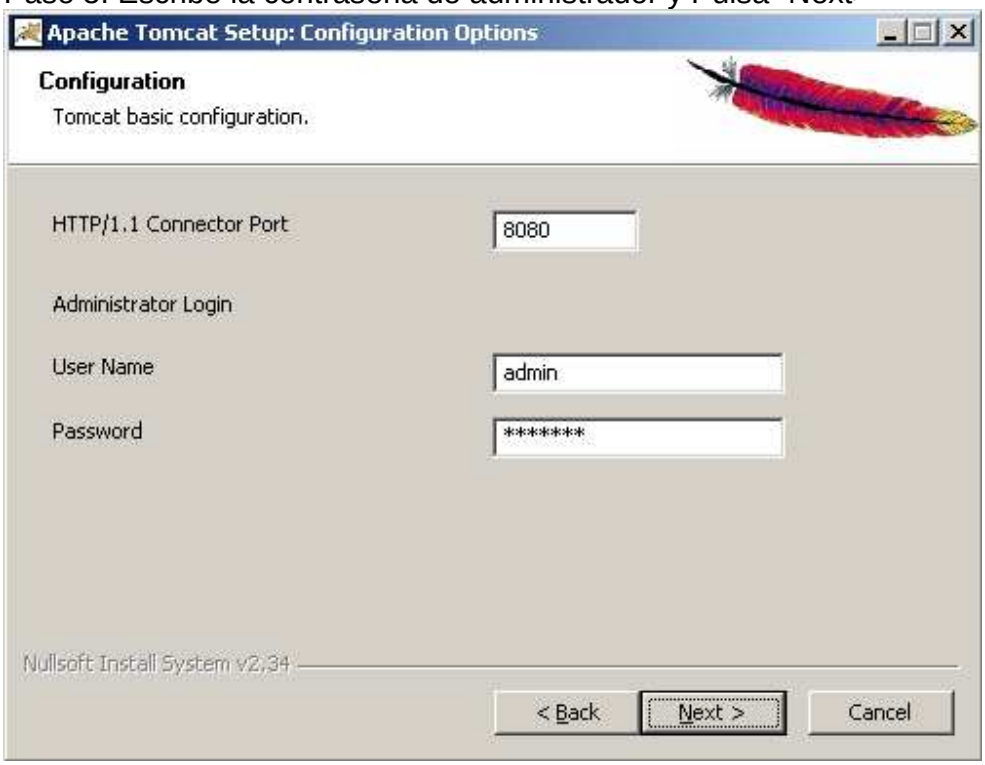


No hay que olvidar marcar el check de “Service Startup” para que el tomcat se instale como un servicio de Windows.

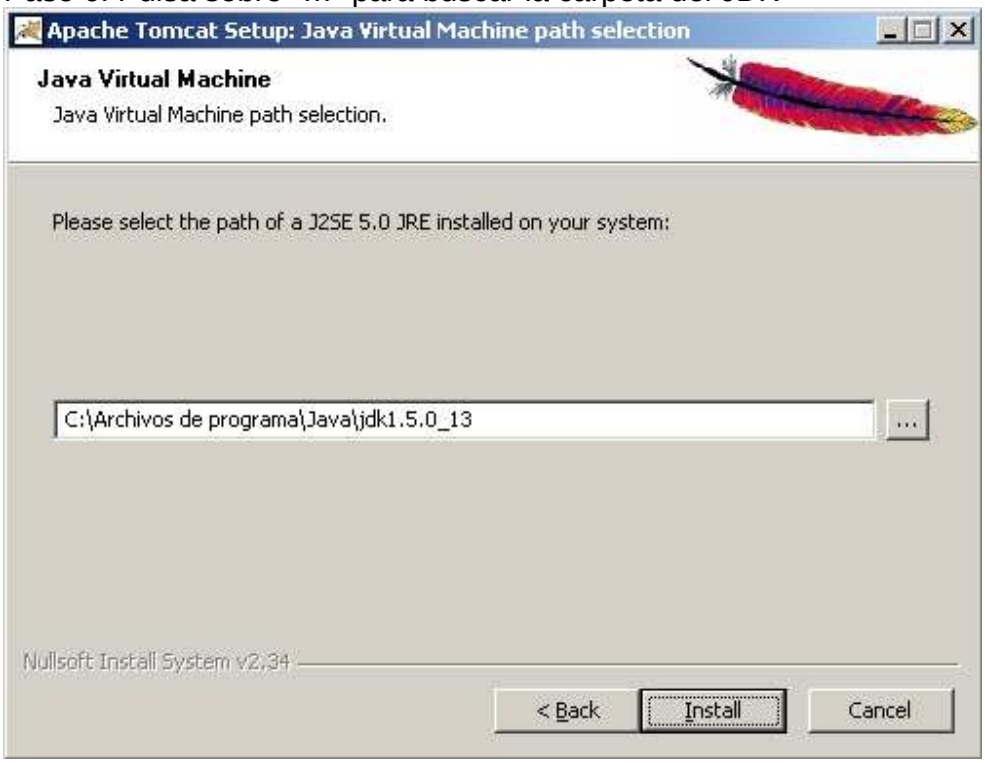
Paso 4:Pulsa “Next>“



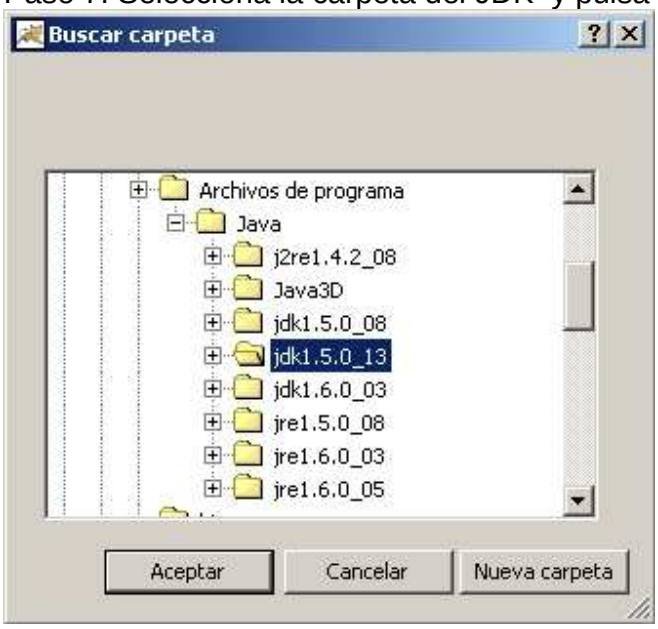
Paso 5: Escribe la contraseña de administrador y Pulsa “Next>“



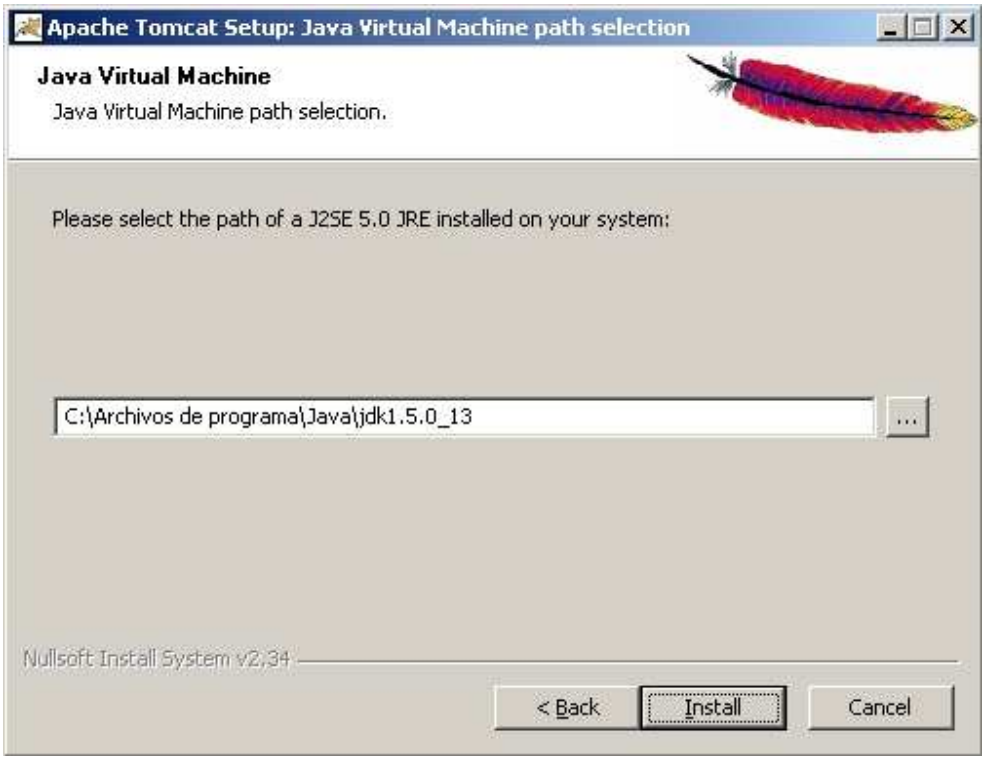
Paso 6: Pulsa sobre ”…” para buscar la carpeta del JDK



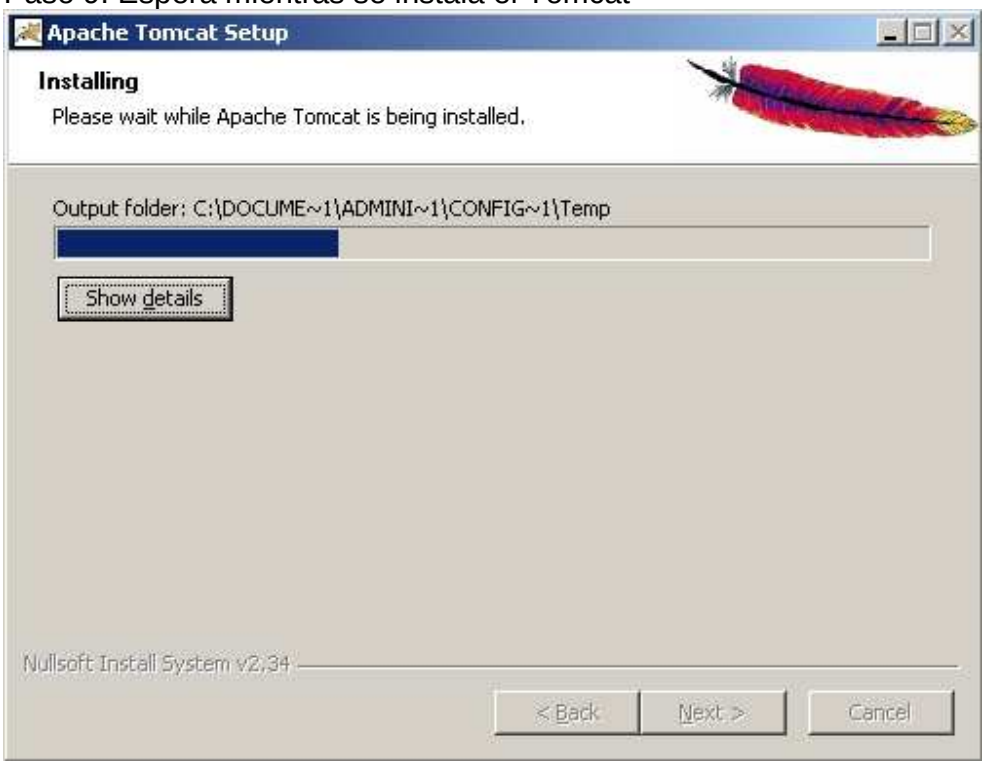
Paso 7: Selecciona la carpeta del JDK y pulsa “Aceptar”



Paso 8: Pulsa “Install”

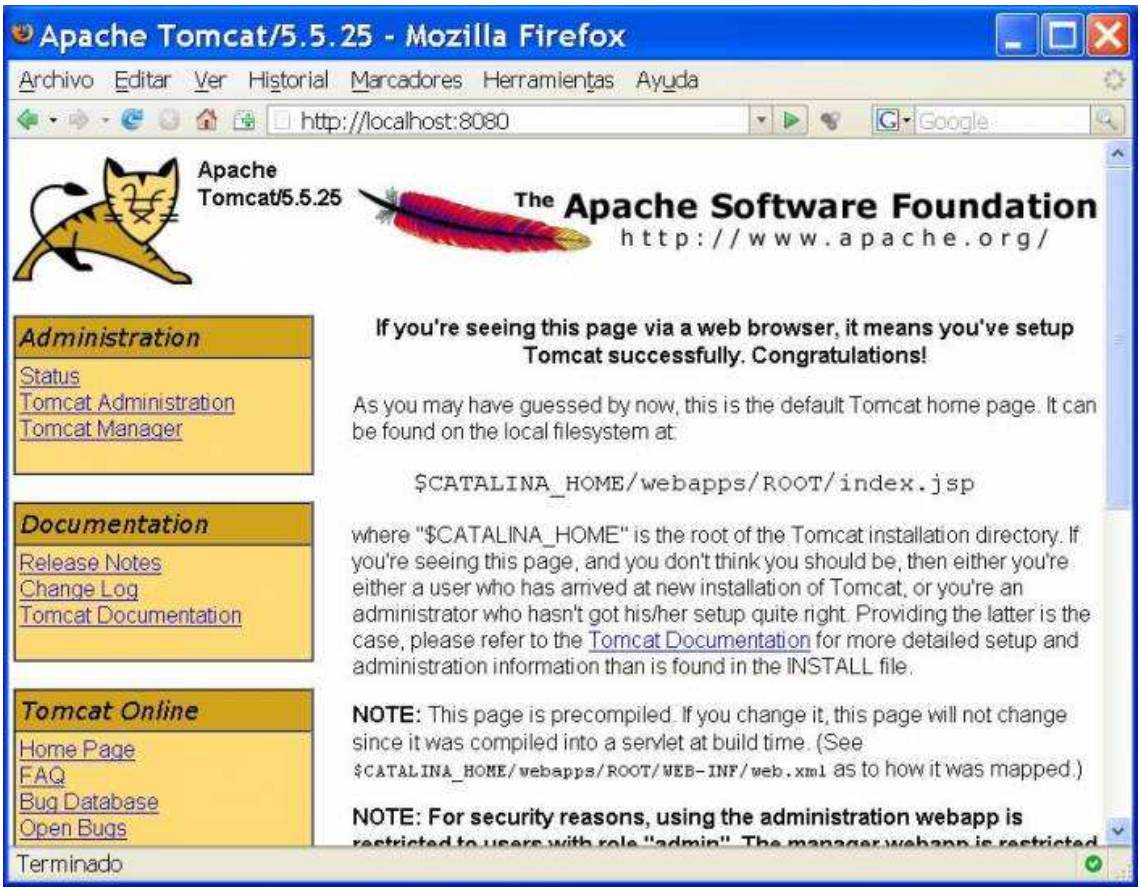


Paso 9: Espera mientras se instala el Tomcat



Paso 10: Pulsa “Finish”

Para comprobar que la instalación del servicio se ha hecho correctamente, desde un navegador se escribe la URL http://localhost:8080 y se vera una pantalla como la siguiente:



Anexo 2 Instalación de Python

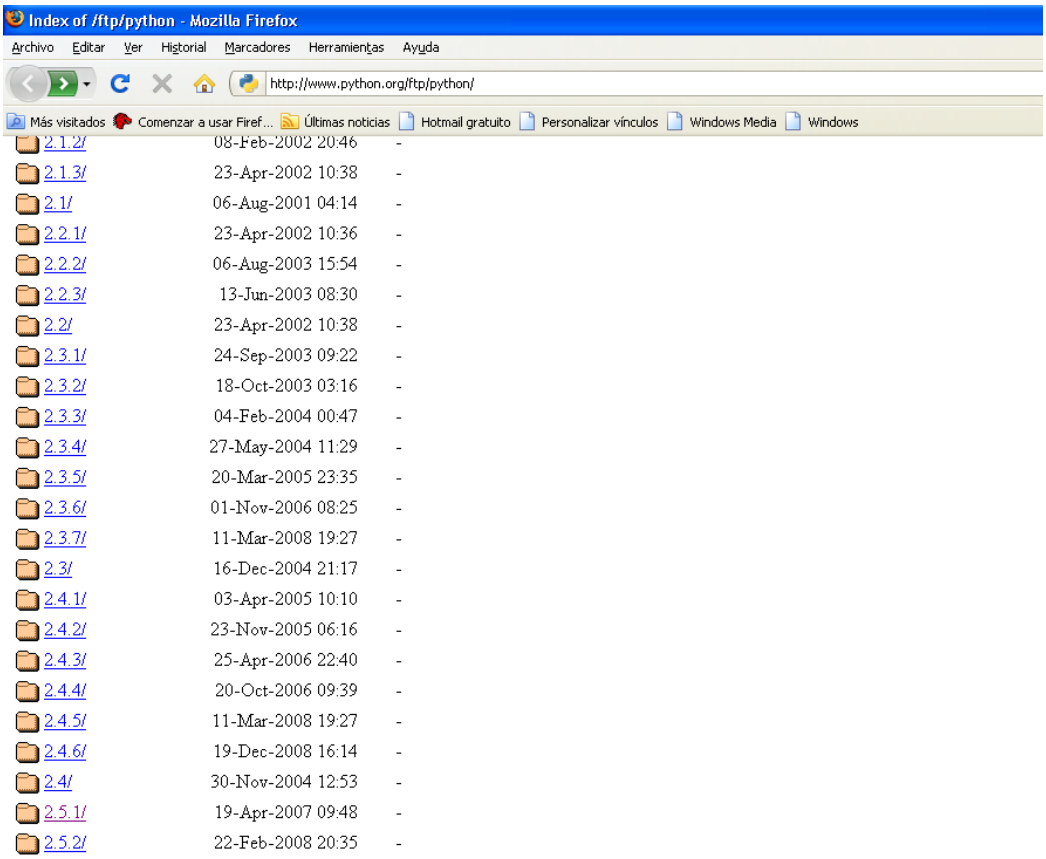
En esta sección se muestra la forma de descargar e instalar Python, el cual es lenguaje de programación.

El proceso de instalación consta de las siguientes tareas:

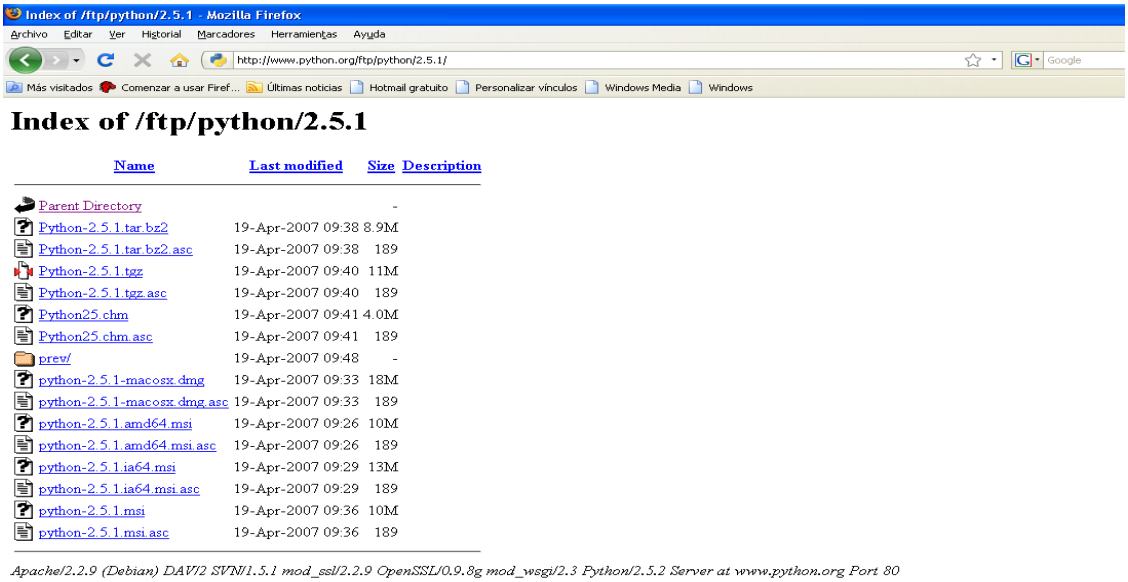
- Descarga
- Instalación

Descarga

1. Descargue el último instalador de Python para Windows de http://www.python.org/ftp/python/ y escogiendo el número de versión más alto que esté en la lista, para descargar el instalador .exe.



2. Haga doble clic en el instalador, Python-2.xxx.yyy.exe. El nombre dependerá de la versión de Python disponible.



Instalación

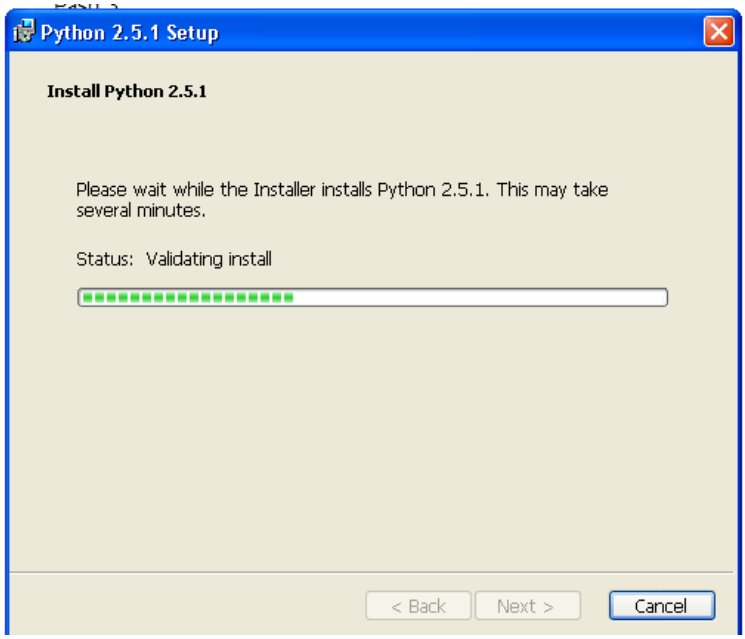
Se ejecutara el programa de instalación apache-tomcat-5.5.25.exe y se llevaran a cabo los siguientes pasos

Paso 1:Pulsa “Next>“

Paso 2: Pulsa “Next>“dejando la opción por default de instalación para todos los usuarios

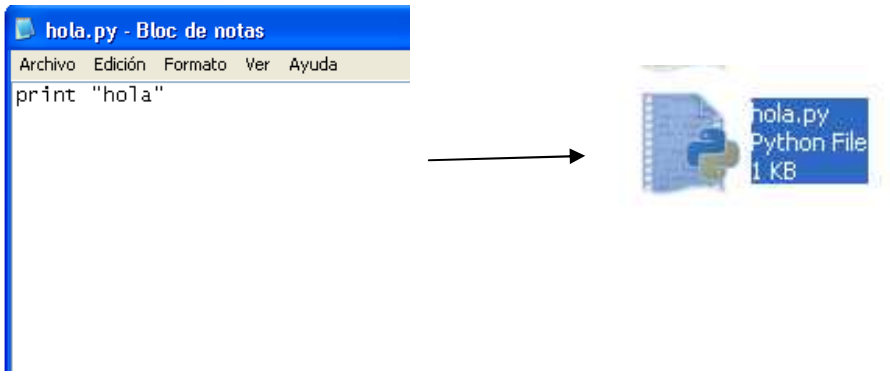
Paso 3: Pulsa “Next>”

Paso 4: esperamos a que finalice el proceso de instalación.



Paso 5: pulsa finish para terminar la instalacion

Para comprobar su correcta instalación se puede crear un archivo con extensión .py el cual cambiara el tipo de icono por uno el cual haga referencia a el lenguaje python.



Referencias

[Budi, et. Al. 1995]

Budi Y., Savio L., Jerry H.,Dik L,1995, A World Wide Web Resource Discovery System*, Proceedings of the Fourth International World Wide Web conference*, FIWWWC, Boston, Dec 1995, pp 145-158

[Clement Y. et al, 1999]

Clement Y., Weiyi M., King-lup L., Wensheng W., Naphtali R.,1999, Efficient and effective metasearch for a large number of text databases, Proceedings of the Eighth International Conference on Information and Knowledge Management, EICIKM, Kansas City, Missouri, United States,1999,pp 217 - 224 .

[Guzman C.,2005]

Guzmán C., Montes y Gomes M., Rosso P.,2005, Búsqueda de colocaciones en la web para sinónimos de Wordnet, Red de Revistas científicas de América Latina y el Caribe, España y Portugal, Mayo-Agosto 2005,vol. 15, número 002, pp 50-56.

[Juárez G.,2007]

Juárez G. A., ”Extracción de respuestas mediante aprendizaje automático utilizando atributos léxicos”, Tesis de Maestría en Ciencias Computacionales, INAOE 2007.

[López O.,2002]

López O.,” Sistema interactivo para la búsqueda de información”, Tesis doctoral en sistemas informáticos, Universidad Nacional de Educación a Distancia, 2002.

[Martínez S.,2002]

Martínez S., F., M. Martín, y L.A.Ureña. 2003. SINAI at CLEF 2002: Experiments with merging strategies. Advances in Cross-Language Information Retrieval. Lecture Notes in Computer Science. Springer Verlag, pp. 187–197.

[Lara N., Pablo M., 2004]

Lara N., Pablo M., 2004, *Agentes inteligentes en la búsqueda y recuperación de información.* Manual. Planeta UOC, Barcelona.

[Merlino S.,2001]

Merlino S. 2001, *Acceso y recuperación de información en la World Wide Web. Análisis de motores de búsqueda y metabuscadores.* Tesis de licenciatura, Universidad Nacional de Mar del Plata. Facultad de Humanidades.

[B. Y. y D.L. Lee., 1997]

B. Y. and D.L. Lee. 1997. Server Ranking for Distributed Text Retrieval Systems on the Internet. In *Proceedings of the Fifth International Conference on Database Systems for Advanced Applications*.

[Z. Wu et, al., 2003]

Z. Wu, V. Raghavan, C. Du, M. Sai C, W. Meng, H. He, and C. Yu. *SE-LEGO: Creating Metasearch Engine on Demand*. ACM SIGIR Conference, Demo paper, pp.464, 2003.

[E. Yom-Tovet al.,2004]

E. Yom-Tov, S. Fine, D. Carmel, A. Darlow, andE. Amitay. Juru at TREC 2004: 2004. Experiments with Prediction of Query Difficulty. In Proceeding of the 13th Text REtrieval Conference (TREC-2004). National Institute of Standards and Technology. NIST.