Universidad de las Ciencias Informáticas

FACULTAD 1

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Título: Empleo de las técnicas de agrupamiento basadas en grafos para la Recuperación de Información.

Autora: Yucel Francis Espinosa

Tutores: Ing. Emilio Suri López
Ing. Yirianni Rivero Escalona

La Habana, Cuba

2013


¨El futuro de nuestra Patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia(...)¨
Fidel Castro.

Declaración De Autoría

Declaro ser la única autora de este trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ___ días del mes de _____ del año 2013.

_____

Yucel Francis Espinosa

_____ _____

Ing. Emilio Suri López Ing. Yirianni Rivero Escalona

Agradecimientos

Agradezco:

A mi Nenucho, a mi Riqui y a mi mamá María Cristina, por ser las personas que me inspiran a seguir adelante cuando siento que no puedo lograr mis objetivos. Además, porque me brindan siempre su apoyo incondicional y cariño.

A mis amigas Dayli, Nurisel, Iliana, Lilianet, Merlyn, Mirelis y Arianne porque en estos cinco años se han convertido en una familia para mí y muy importante, por soportar mi extraño carácter.

A mis amigos los jimaguas Yasiel y Yadiel, Daniel, Oscar, Antonio y Ricardo porque ellos siempre me ayudaron cuando tenía una duda.

A la familia de Nurisel, su mamá Olga, Jorge, Cayo y Paulina por sus atenciones y preocupación.

A las personas que me han ayudado a llegar aquí, a Liuba, Ernesto, Reinier, Chuchi y mis tutores, especialmente Yirianni.

A todos los que se preocupan por mí, gracias.

Dedicatoria

A la persona que más amo en el mundo, mi mamá.

Resumen

Para realizar la Recuperación de Información es de gran utilidad el agrupamiento de documentos. Este trabajo realiza un estudio de las técnicas de agrupamiento basadas en grafos. Además, utiliza una función de semejanza simétrica para documentos y un grupo de herramientas que permiten desarrollar estas técnicas en un módulo llamado “algoritmos” para su uso en el CMS Drupal, versión 7. El objetivo del agrupamiento es lograr que cuando se realice una búsqueda los documentos se encuentren agrupados por su similitud. Los algoritmos implementados son GLC, Compacto Incremental y Fuertemente Compacto Incremental.

Palabras claves: algoritmo Compacto Incremental, algoritmo Fuertemente Compacto Incremental, algoritmo GLC y Recuperación de Información.

Índice

Introducción

A medida que las tecnologías computacionales progresan, junto a ellas aumenta la esfera de la información digital. La necesidad de la utilización de herramientas para mostrar y organizar el elevado volumen de información de la colección, se hace cada día más esencial, para así permitir y facilitar a los usuarios la búsqueda y utilización de la información (CASTILLO, 2010). Con el objetivo de solucionar la anterior problemática surge la creación de una ciencia relacionada con la búsqueda, conocida como “Recuperación de Información” (RI), la cual es la encargada de buscar y ofrecer información relacionada con ciertas palabras claves escritas por el usuario mediante una consulta.

En la RI las principales actividades que se realizan son la indización y la búsqueda; cuando se habla de indización se refiere a la representación y descripción de la información. Por otra parte, la RI es un proceso de comunicación que tiene dos elementos de comunicación, usuario-intermediario (BÄR, 2012). Este proceso comienza con la necesidad de información de un usuario que acude a un intermediario a través de una consulta en lenguaje natural. Posteriormente, el segundo elemento traduce la consulta a una estrategia de búsqueda.

Para la RI intervienen los soportes electromagnéticos, especialmente las bases de datos. Actualmente cuando se habla de RI se da por hecho que interviene un sistema informático, es decir, un sistema de RI, contenido por una serie de programas que se ocupan de gestionar la información y las consultas realizadas. Este sistema es un dispositivo interpuesto entre un usuario y la colección de información, con el objetivo de recuperar la información que es relevante para la consulta del usuario y retener aquella que no lo es (BÄR, 2012).

Para llevar a cabo la RI existe una serie de técnicas, entre ellas se encuentran las técnicas de agrupamiento. El objetivo de las mismas es obtener grupos o conjuntos entre los elementos, de tal manera que los asignados al mismo sean similares (PONS, 2004). Dicho agrupamiento permite que el resultado de la RI sea más ajustado a la necesidad de información y más rápido al reducirse el dominio de búsqueda.

Introducción

Para realizar la RI en el Sistema de Gestión de Contenidos (CMS por sus siglas en inglés, *Content Management System*) Drupal en su versión 7 se realiza mediante índices invertidos. Lo que hacen los motores de búsqueda es analizar las páginas y guardar la información relevante para agilizar el proceso de búsqueda posterior. Drupal mantiene un índice de los contenidos del sitio web (GIL, 2012), lo que provoca que sea errónea la RI en casos donde la cantidad de información sea muy elevada y que no todas las búsquedas cumplan con las expectativas del usuario. Al no hacer uso del agrupamiento de la información teniendo en cuenta la distribución por diferentes temáticas, puede dar paso a que no exista una buena precisión en la respuesta. Por otra parte, al no estar agrupados influye en la velocidad de respuesta.

Debido a lo anteriormente descrito se tiene el siguiente problema de la investigación: ¿Cómo utilizar el agrupamiento para mejorar el proceso de la RI, en el CMS Drupal, versión 7? Se declara como objeto de estudio: Técnicas de agrupamiento y como campo de acción: Técnicas de agrupamiento basadas en grafos.

Para dar respuesta al problema planteado se define como objetivo general: Desarrollar un módulo para el CMS Drupal en su versión 7, que implemente las técnicas de agrupamiento basadas en grafos GLC, Compacto Incremental y Fuertemente Compacto Incremental para mejorar el proceso de RI.

Dando paso a los objetivos específicos siguientes:

- Resumir los aspectos teóricos relacionados con las técnicas de agrupamiento para la RI.
- Desarrollar un módulo que posibilite la aplicación de técnicas de agrupamiento basadas en grafos.
- Realizar las pruebas para validar el módulo implementado.

Para dar cumplimiento a los objetivos planteados se definen las siguientes tareas de la investigación:

- Conceptualización de los aspectos teóricos.

Introducción

- Descripción de las técnicas de agrupamiento para la RI.
- Caracterización de las técnicas de agrupamiento basadas en grafos para la RI.
- Descripción de las herramientas y tecnologías a utilizar.
- Definición de la propuesta de solución donde se implementan las técnicas de agrupamiento basadas en grafos para la RI.
- Enunciación del modelo de dominio.
- Definición de los requerimientos.
- Visión de la arquitectura del módulo.
- Implementación del módulo donde se desarrollan las técnicas de agrupamiento basadas en grafos para la RI.
- Realización de las pruebas de precisión y exhaustividad al módulo implementado.
- Evaluación del resultado del proceso de prueba**.**

Para llevar a cabo la investigación se hace uso de un grupo de métodos científicos:

Analítico-Sintético: Se utilizó para el análisis de los subdominios dentro del objeto de estudio y su posterior integración, de esta forma, se facilita la comprensión de la problemática. Se usó fundamentalmente para el entendimiento de la hipótesis del agrupamiento y en el estudio de las técnicas de agrupamiento.

Inductivo-Deductivo: Permitió a partir del estudio de casos de agrupamiento en diversos escenarios, sacar conclusiones de cómo aplicarlo en la arquitectura del CMS Drupal, versión 7.

Histórico-Lógico: Se empleó en el estudio de las técnicas de agrupamiento y particularmente en el análisis de la evolución de las basadas en grafo.

Introducción

Constatación: Permitió verificar la efectividad del resultado obtenido a partir de confrontarlo con lo que debe ser la respuesta ideal del sistema.

El presente documento está estructurado en los capítulos que a continuación se muestran:

Capítulo 1: Marco teórico conceptual.

Estudio de los aspectos teóricos fundamentales para el entendimiento del trabajo. Se mencionan ejemplos de aplicaciones que hacen uso de las técnicas de agrupamiento GLC, Compacto Incremental y Fuertemente Compacto Incremental. Además, se mencionan las características de las herramientas y el lenguaje a utilizar.

Capítulo 2: Características de la solución para la Recuperación de Información.

Descripción de la propuesta de solución. Se exponen además los diagramas necesarios para un mejor entendimiento de la solución y los requerimientos que debe cumplir.

Capítulo 3: Implementación y prueba

Descripción de los algoritmos de agrupamiento basados en grafos GLC, Compacto Incremental y Fuertemente Compacto Incremental y del modelo vectorial similitud mediante el coseno. Contempla además el proceso de pruebas del módulo desarrollado.

Capítulo 1: Marco teórico conceptual.

El objetivo de este capítulo es presentar un estudio de los aspectos teóricos que facilitan el entendimiento de la investigación. Se mencionan ejemplos de las aplicaciones existentes sobre los algoritmos de agrupamiento GLC, Compacto Incremental y Fuertemente Compacto Incremental. Además, se caracterizan las herramientas y el lenguaje de programación que se utilizan para el desarrollo.

1 Conceptos fundamentales

1.1 Información

En el contexto de la gestión o dirección de una organización, se asume como información el conjunto de datos interrelacionados que aportan algún beneficio. Para que ella exista debe contener al menos las tres dimensiones básicas que la definen: las cuales son, contexto, concepto y valor. Para trasmitir adecuadamente la definición se presenta el siguiente ejemplo.

La empresa cobró la cuenta. ¿Qué cuenta?

Cobró 10 000 CUC. ¿Quién, qué?

Empresa X, 10 000 CUC. ¿Qué?

Como se puede apreciar la omisión de cualquier parte de la información implica la pérdida de capacidad de entendimiento, lo que supone un dato fuera de contexto y sin aplicación o valor cognoscitivo.

En el diccionario “Gestión del conocimiento e informática” se define información como todo lo que un computador proporciona como salida después de un determinado proceso en el que ha actuado sobre una materia prima. Esta definición da lugar a que determinados elementos puedan ser considerados como informaciones de salida por unos y datos de entrada por otros. Así, por ejemplo, la cinta con las declaraciones de los ingresos y de las retenciones de los

asalariados de una determinada institución, son informaciones de salida para el centro informático de esa institución, mientras que esa misma cinta, para la Agencia Tributaria, tiene la consideración de datos de entrada a su sistema (MESTRE, 2000).

1.2 Recuperación de Información

La RI abarca los aspectos intelectuales de la descripción de información y su especificación para la búsqueda, y también cualquier sistema, técnica o máquina que se utilice para llevar a cabo la operación (MOOERS, 1951).

En los términos de la presente investigación se asume RI como el conjunto de operaciones, que recupera la información contenida en una colección de datos asociada a los criterios de búsqueda definidos por el usuario y la pone a su disposición para su posterior procesamiento.

Entre los tópicos de RI se encuentran la clasificación, sumarización, modelos de recuperación y el agrupamiento. Siendo este último el que se utilice en el presente trabajo (TOLOSA, 2007).

1.3 Algoritmo de agrupamiento

El agrupamiento tiene como propósito crear grupos de objetos de manera tal que los objetos pertenecientes a un grupo tengan entre ellos máxima semejanza y con respecto a los objetos pertenecientes a otros grupos, tengan entre ellos menor semejanza (PONS, 2004).

El algoritmo de agrupamiento es empleado en diferentes contextos como por ejemplo, en compresión de datos, segmentación de imágenes y en RI. Existen diversos algoritmos de agrupamiento que se clasifican dependiendo de algunos criterios para el uso de las distintas aplicaciones, como los basados en optimización, los jerárquicos divisivos y los basados en grafos. El presente trabajo se desarrolla basándose en este último y específicamente en los algoritmos GLC, Compacto Incremental y Fuertemente Compacto Incremental.

1.4 Algoritmos de agrupamiento basados en grafos

Los algoritmos de agrupamiento de documentos basados en grafos arrojan como resultado final un grafo G (V, E, w), que depende de la semejanza entre los objetos. Dicho grafo se encuentra

constituido por los vértices (V) que representan los documentos de una colección y el peso (w) de cada arista (E) se encuentra representado por la semejanza entre un par de documentos (PÉREZ, 2008).

Grafo de semejanza

El grafo de semejanza es un grafo G (V, E, w) donde los vértices (V) representan los documentos de la colección y las aristas (E) están etiquetadas por la semejanza entre un par de documentos. Cuya semejanza (w) es definida por una función de semejanza simétrica entre documentos (PÉREZ, 2008).

Grafo de β-semejanza

El grafo de β-semejanza se encuentra representado por $G_\beta = (V, E_\beta)$, en el cual se eliminan todas las aristas que cumplan que $w(E_{i,j}) < \beta$. Siendo β es un umbral de semejanza, que cumple 0< =β= >1 (PÉREZ, 2008).

Grafo de máxima β-semejanza

El grafo de máxima semejanza $G_{max} = (V, E)$ es un grafo dirigido donde las aristas cumplen que $w(E_{i,j}) =$ máximo $w(E_{i,n})$ (PÉREZ, 2008).

Componente fuertemente conexa

Una componente fuertemente conexa de un grafo orientado es un conjunto de vértices en el cual para todo vértice en el conjunto existe un camino a cualquier otro vértice del conjunto (PONS, 2004).

Grafo reducido según la fuerte-conexidad

Sean G = (V, E) un grafo orientado y P una partición del conjunto de vértices V, donde X_i denota a cada elemento de P. Se llama grafo reducido de G, y se expresa como $G_r = (X, E_r)$, al grafo cuyos vértices son los subgrafos generados por los elementos X_i de P y existe un arco del vértice x_i al x_j si existe un vértice v en X_i y un vértice y en X_j tal que existe el arco de v a y en G (KAKES,

1987).

Base del grafo

Sea G = (V, E) un grafo orientado y B ⊂ V. B es una base del grafo G si se cumplen las dos condiciones siguientes (KAKES, 1987):

- Todo vértice del conjunto V es descendiente de al menos un vértice de B.
- No existe en el conjunto V un subconjunto de menor número de elementos con dicha propiedad.

Ejemplo del empleo de las últimas tres definiciones:

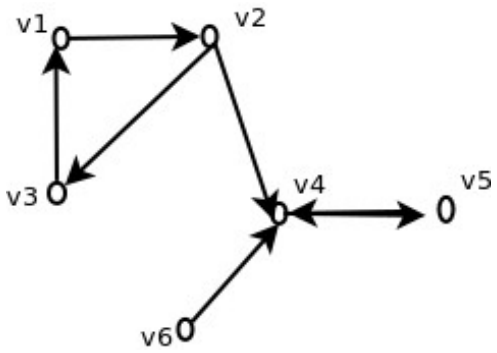

Figura 1: Grafo G

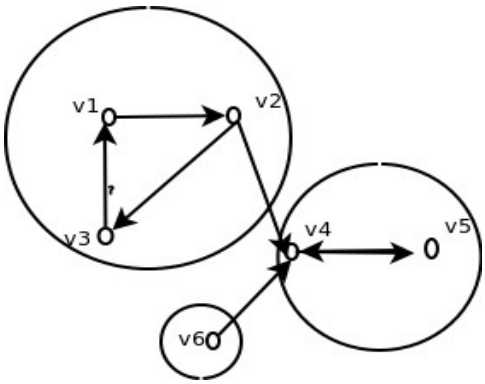

Figura 2: Componentes fuertemente conexas de G

Las componentes fuertemente conexas:

son las que se encuentran encerradas en círculos.

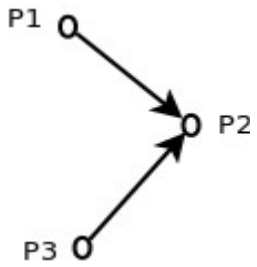

Figura 3: Grafo Gr

Grafo reducido según la fuerte-conexidad,
donde P_1 ={v_1, v_2 y v_3}, P_2 ={v_4 y v_5} y P_3 ={v_6}

Las bases del grafo son {v_1, v_6}, {v_2, v_6} y {v_3, v_6}

Algoritmo basado en grafo GLC

El algoritmo basado en grafo GLC analiza la semejanza de cada documento de la colección una sola vez. Definiendo para el documento que esté analizando los documentos de su grafo de β-semejanza. En caso de que no exista ningún documento en el grafo de β-semejanza, el documento que se está analizando en ese momento va a formar un grupo. Mientras que, en caso contrario, se crea un grupo formado por la unión del documento analizado y todos los grupos a los que pertenezcan los documentos que se encuentran en el grafo de β-semejanza (PÉREZ, 2008).

Algoritmo basado en grafo Compacto Incremental

El algoritmo Compacto Incremental dado el grafo de máxima semejanza G_{max} = (V, E) que representa la colección de documentos, obtiene de forma incremental un conjunto de grupos disjuntos G = {G_1, G_2, ..., G_K} en el que cada G_i es un conjunto compacto. Los conjuntos compactos coinciden con las componentes conexas del grafo G_{max} (PÉREZ, 2008).

A la llegada de un nuevo documento se analiza su semejanza con cada uno de los documentos

pertenecientes a los grupos compactos. En caso que no exista ninguna semejanza el nuevo objeto que se está analizando va a formar parte de un nuevo grupo compacto. En caso de que sí existan semejanzas, los objetos conectados a él son eliminados de los grupos compactos a los que pertenecían y lo cual puede traer consigo que los documentos que queden se encuentren inconexos.

Debido a lo anteriormente planteado se pasa a realizar la reconstrucción de los grupos compactos que poseen objetos semejantes con el actual objeto, los cuales se distribuyen en los siguientes grupos:

Grupos a procesar: Pertenece a este grupo si cumple con algunas de las condiciones:

- El objeto O tiene como más β-semejante al objeto nuevo y los objetos al que O se conectaba dejan de serlo. Además, O no es el más semejante a ninguno del resto del grupo compacto.
- El objeto O tiene como más β-semejante al objeto nuevo y el o los objetos al que O se conectaba dejan de serlo. Además, O es el más semejante al menos uno del resto del grupo compacto.

Objetos a unir:

- El objeto O tiene como más β-semejante al objeto nuevo y el único objeto al que O se conectaba deja de serlo. Además, O no es el más semejante a ninguno del resto del grupo compacto.

Grupos a unir: Pertenece a este grupo si no se encuentra en grupos a procesar y además cumple con una de las condiciones siguientes:

- O es el más β-semejante del nuevo objeto.
- El nuevo objeto se encuentra conectado con O y no se rompe ninguna arista del grafo de máxima β-semejanza.

Todos los objetos que forman parte de los grupos anteriores junto al objeto nuevo construyen un nuevo grupo compacto (GARCÍA, 2005).

Algoritmo basado en grafo Fuertemente Compacto Incremental

El algoritmo Fuertemente Compacto Incremental representa la colección de documentos a través del grafo de máxima semejanza G_{max} = (V, E), formando de manera incremental conjuntos fuertemente compactos (CFC). A la llegada de un nuevo documento se calcula la semejanza con el resto de los documentos que forman parte de los CFC. Si no existe ninguna semejanza, el documento que se analiza es un conjunto fuertemente compacto. En caso contrario se crea un conjunto compacto con el objeto nuevo y los conectados a él. Luego se reconstruyen los conjuntos compactos a partir de los conjuntos fuertemente compactos que perdieron conexiones producto de la llegada del nuevo documento. A partir de estos se construyen los cubrimientos de los CFC. Haciendo uso de las definiciones de componente fuertemente conexa, grafo reducido según la fuerte-conexidad y de base de un grafo se crean los CFC (PONS, 2004).

La semejanza entre documentos es un factor indispensable en estos algoritmos de agrupamiento. Para lograr un valor de semejanza entre los documentos es muy usado en operaciones de la RI el modelo vectorial mediante coseno. El modelo anteriormente mencionado es el que se emplea para el desarrollo de dichos algoritmos y a continuación se refleja una breve explicación del modelo vectorial.

1.5 Modelo vectorial: similitud mediante el coseno

El modelo vectorial representa un documento x en forma de un vector, el cual representa la frecuencia de términos en dicho documento. El modelo vectorial mediante el coseno está definido por la expresión:

$$\text{CosSim}(d_j, d_q) = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^{2} \cdot \sum_{i=1}^{t} w_{iq}^{2}}}$$

w_{ix} es el peso del término i en el documento x, por lo que un documento puede tener w_{nx}, n pesos dependiendo de la cantidad de términos que se esté analizando (ZAZO, 2002).

2 Empleo de los algoritmos basados en grafos GLC, Compacto Incremental y Fuertemente Compacto Incremental

Con la realización de un estudio sobre el uso de algoritmos basados en grafos, se obtiene que estos han sido empleados en diferentes trabajos científicos como son:

Tesis doctoral “Algoritmos de agrupamientos basados sobre grafos y su paralelización”, por el autor Reynaldo José Gil García. Dicha tesis propone la paralelización de los algoritmos de agrupamiento basados en grafos, con el objetivo de que sea la mejor opción para crear grupos que tengan calidad ante un gran volumen de información o cuando el algoritmo de agrupamiento que se desee utilizar tenga una complejidad computacional muy elevada.

En XIV Jornadas de paralelismo el trabajo “Algoritmo paralelo para detectar sucesos en noticias en líneas” realizado por Reynaldo José Gil García, José Manuel Badía Contelles y Aurora Pons Porrata hace uso del algoritmo Compacto Incremental. El objetivo de este trabajo es exponer un algoritmo paralelo que detecte los sucesos de noticias en líneas.

Tesis doctoral “Desarrollo de algoritmos para la estructuración dinámica de información y su aplicación a la detección de sucesos” por la autora Aurora Pons Porrata. El objetivo de dicha tesis es identificar en un flujo continuo de noticias, aquellas que pertenecen a un nuevo tópico o a uno conocido previamente.

En la realización de un proceso de desarrollo de *software* se tienen en cuenta un conjunto de herramientas y lenguajes los cuales hacen posible que se desarrolle con éxito lo que se desea. A continuación se realiza una caracterización de los seleccionados para dar cumplimiento al objetivo general.

Capítulo 1: Marco Teórico Conceptual.

3 Herramientas

3.1 Entorno de Desarrollo Integrado (IDE)

IDE por sus siglas en inglés *Integrated Development Environment* es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (MARRERO, 2012). NetBeans es un ejemplo de este tipo de entorno y a continuación se muestran sus características.

NetBeans 7.01

NetBeans es de código abierto, su plataforma puede ser usada para desarrollar cualquier tipo de aplicación. Se puede hacer reutilización de módulos. Instalación y actualización simple. Incluye *Template* y *Wizards* y posee soporte para PHP (GIMÉNEZ, 2012).

3.2 Sistema de Gestión de Contenidos

Traducido al inglés como *Content Management System* (CMS) es una herramienta que permite hacer un portal web. Los CMS son un sitio web que permite una sencilla interacción al usuario y con su uso se puede realizar la gestión de contenido de la página, ejemplo crear, publicar y eliminar el contenido.

La ventaja principal del uso de CMS es que sin necesidad de tener conocimiento de programación ni de maquetación se puede diseñar un portal web dinámico, pues el CMS aporta todas las opciones de diseño y el usuario administrador únicamente debe definir los contenidos de cada apartado de su web. Son generalmente gratuitos, se pueden descargar desde las páginas oficiales de cada CMS. Además de que su uso no implica ninguna condición ni restricción específica que limite el funcionamiento.

También es importante mencionar que por medio del uso de CMS se agilizan todas las tareas relacionadas con actualización, *backup* y reestructuración de los portales web, ya que toda la información importante está almacenada en la base de datos vinculada al portal (ROMEO, 2012).

Capítulo 1: Marco Teórico Conceptual.

Enmarcado en la familia de CMS se encuentra Drupal, el cual posee suficiente documentación y una gran comunidad activa. Hace uso de gestores de bases de datos como PostgreSQL, MySQL y SQLite los cuales se encuentran bajo la licencia GNU/GPL. Además, Drupal es de código abierto. Es un CMS que tiene una potente y sencilla API de programación de nuevas funciones.

Los CMS trabajan con base de datos por lo que se hace necesario el estudio de los sistemas de gestión de base de datos para seleccionar cuál utilizar.

3.3 Sistema de gestión de base de datos

Sistema de gestión de base de datos (SGBD) es una estructura específica para el almacenamiento de datos en un sistema de información automatizado, que posee las siguientes características (MESTRE, 2000):

- Los programas de tratamiento de los datos y los propios datos son totalmente independientes.
- Son sistemas multiusuario y multiprogramación.
- Permiten el acceso directo por todos los campos que se desee.
- Evitan la redundancia de la información.
- Son muy flexibles en la modificación de su diseño.
- Disponen de lenguajes de consulta muy sencillos de utilizar.

Los SGBD que Drupal utiliza son PostgreSQL, SQLite y MySQL. Se selecciona MySQL pues es el sistema de bases de datos en el cual se encuentra la colección de documentos que se agrupa como ejemplo para ver el funcionamiento de los algoritmos. A continuación se muestran las características del SGBD seleccionado:

MySQL 5.5.29

MySQL es un SGBD relacional, multiusuario y de código abierto. La principal función de este sistema es la velocidad y la robustez, para las columnas soporta gran cantidad de tipos de datos. Además, funciona sobre múltiples plataformas y sistemas operativos. Presenta un excelente nivel de seguridad en los datos y es fácil de configurar e instalar. Por su implementación multihilo, tiene un sistema flexible de gestión de usuarios y contraseñas (VALDIVIA, 2012).

3.4 Herramienta CASE para el modelado visual

Las herramientas CASE *Computer Aided Assisted Autimated Software Systems Enginieering* para el Lenguaje Unificado de Modelado (UML) se definen como programas que ayudan a los miembros del equipo de desarrollo de *software* a automatizar el proyecto, durante el ciclo de vida de un *software*.

Las herramientas CASE tienen el propósito de dar cumplimiento a diferentes objetivos (MARTÍNEZ, 2012):

- Mejorar la productividad en el desarrollo y mantenimiento del *software*.
- Aumentar la calidad del *software*.
- Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del *software*, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayuda a la reutilización del *software*, portabilidad y estandarización de la documentación.

- Gestión global en todas las fases de desarrollo de *software* con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del *software*.

Entre las más usadas se encuentran Rational Rose y Visual Paradigm. Se selecciona Visual Paradigm por la estabilidad de ejecución en diferentes sistemas operativos. Además, posee una licencia libre y comercial. Puede generar código a partir de los modelos y viceversa. Emplea las últimas notaciones de UML, ingeniería inversa, generación de código, entre otros. También soporta aplicaciones web y exporta en formato HTML (TORRES, 2012).

A continuación se define el lenguaje de programación a utilizar teniendo en cuenta el trabajo con el CMS Drupal.

4 Lenguaje de programación asociado a Drupal

Lenguaje de programación es un lenguaje especial, no natural, diseñado con un vocabulario, morfología y sintaxis muy simples y rígidas y orientado a la programación de instrucciones elementales cuya ejecución por un determinado sistema físico da lugar a la realización de una tarea (MESTRE, 2000).

Para el desarrollo de este trabajo se selecciona el lenguaje de programación PHP, que representa las siglas de *Hypertext Preprocessor*, el cual es gratuito e independiente de plataforma, con una gran librería de funciones y mucha documentación. Está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de desarrolladores. Soporta en cierta medida la orientación a objeto. Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros. No requiere definición de tipos de variables ni manejo detallado del bajo nivel. Finalmente, posee una capacidad de expandir su potencial utilizando módulos (PACHECO, 2012).

Capítulo 1: Marco Teórico Conceptual.

Conclusiones parciales

El estudio de los conceptos y bases teóricas ayudaron a lograr una mejor comprensión del tema a los lectores. Además, se seleccionaron las herramientas NetBeans, el CMS Drupal, PostgreSQL, Visual Paradigm y el lenguaje PHP para desarrollar el módulo, el cual contendrá los algoritmos GLC, Compacto Incremental y Fuertemente Compacto Incremental.

Capítulo 2: Características de la solución para la Recuperación de Información.

En el presente capítulo se muestra el modelo de dominio y los requerimientos definidos para lograr el objetivo general. Además, se realiza la propuesta de solución, se expone la arquitectura que presenta el CMS Drupal 7 por la cual se guía el desarrollo de la solución y el modelo de datos.

1.1 Modelo de dominio

El modelo de dominio ofrece un mejor entendimiento en el área donde se esté trabajando. Dicho modelo está compuesto por las clases más importantes dentro del contexto y va de lo más general a lo más específico.

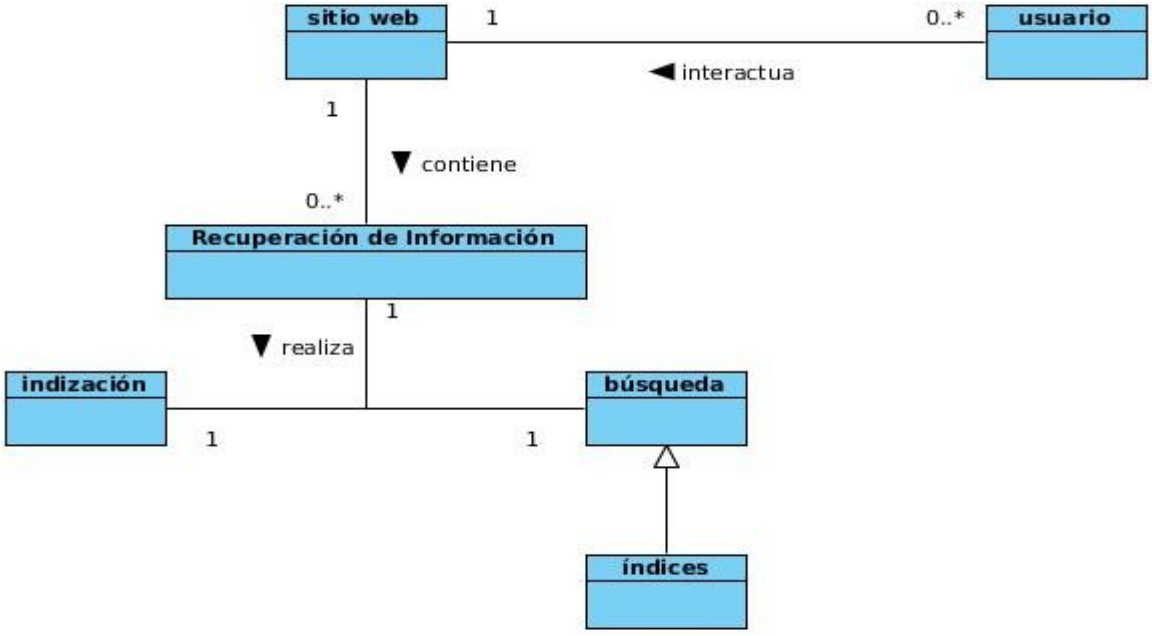

Figura 4: Modelo de domino

Capítulo 2: Características De La Solución Para La Recuperación De Información.

Definición de las clases del modelo de dominio:

Usuarios: Son todos aquellos usuarios que tienen el privilegio de interactuar con el servicio buscar del sitio web.

Sitio web: Es un sitio que posee varias páginas web y pone a disposición de los usuarios un conjunto de servicios.

RI: Es el paso por la que se determina las necesidades de información.

Indización y búsqueda: Son procesos que utiliza la RI.

1.2 Especificación de requerimientos

Para cumplir con el objetivo general se definieron los requerimientos que más adelante se describen:

- Definir la similitud entre documentos: Calcula la semejanza entre un par de documentos.

- Crear grupo a procesar: Forma un grupo con los objetos que poseen las características asociadas a este grupo.

- Crear objeto a unir: Forma un grupo con los objetos que poseen las características asociadas a este grupo.

- Crear grupo a unir: Forma un grupo con los objetos que poseen las características asociadas a este grupo.

- Agrupar los documentos: Tiene como propósito realizar la construcción del grafo con los documentos de la colección. Dicha construcción depende del algoritmo a utilizar pues son tres GLC, Compacto Incremental y Fuertemente Compacto Incremental. Por lo que se

obtiene como resultado final tres grafos.

1.3 Descripción de la propuesta de solución

El módulo implementa los algoritmos GLC, Compacto Incremental y Fuertemente Compacto Incremental con el fin de lograr el agrupamiento de los documentos por tres formas distintas dentro de una colección de documentos, para así enriquecer la RI en el CMS Drupal versión 7.

Un primer paso para llevar a cabo cada uno de estos algoritmos es partir de una colección de documentos previamente indizados, para así tener una representación mediante las principales palabras que componen a cada documento. Además, implementar una función de semejanza que permita conocer la similitud que presenta un documento con otro. La función que se utiliza es el coseno del ángulo entre dos vectores conocido como modelo vectorial mediante coseno. Luego de contar con lo anteriormente mencionado se pasa a realizar el grafo de β-semejanza dependiendo del algoritmo. Donde se recorren todos los documentos y se va formando un grafo tratando siempre que la arista que relaciona un vértice con otro sea la que mayor peso tenga entre ambos. En el algoritmo GLC el peso se define como 0 ó 1, donde 1 significa que existe semejanza entre un par de documentos. En los casos de Compacto Incremental y Fuertemente Compacto Incremental la semejanza va a estar representada por un número entre 0 y 1, dicho número cuanto más cercano se encuentre del 1 más semejanza tienen los documentos.

1.4 Arquitectura de Drupal 7

En la Figura 5 se encuentran reflejados los componentes por los que Drupal 7 está formado:

Capítulo 2: Características De La Solución Para La Recuperación De Información.

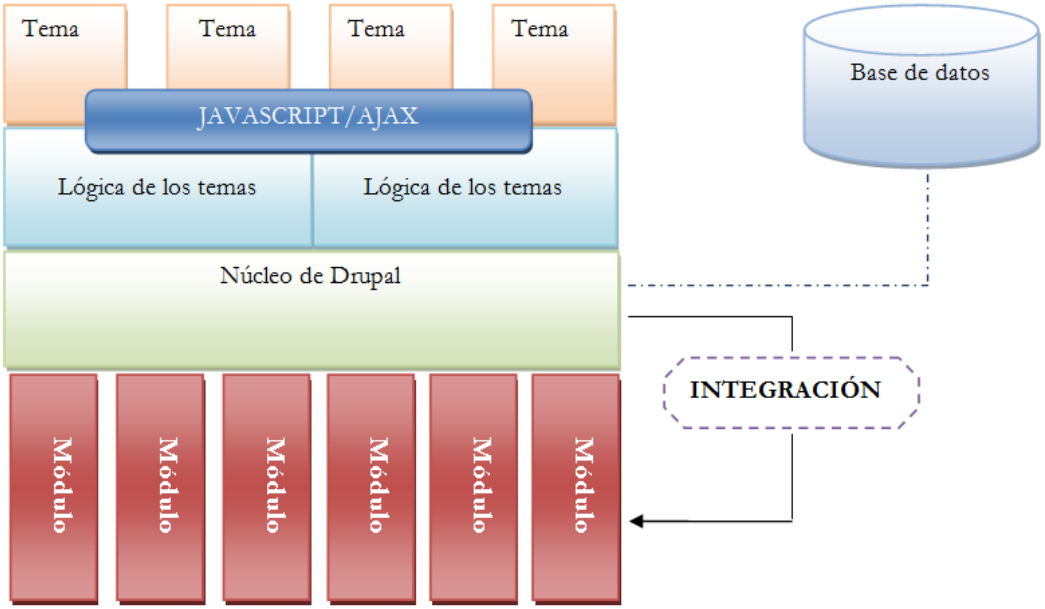

Figura 5: Componentes que forman la arquitectura de Drupal 7 (GIL, 2012).

El código que constituye el núcleo de Drupal está formado por un conjunto de librerías que permiten gestionar los procesos de arranque del sistema. Estas librerías ofrecen además un conjunto de servicios que permiten integrar las funcionalidades adicionales de los módulos, servicios como conexión y administración de la base de datos, gestión de procesos de mailing, tratamiento de imágenes, internacionalización, soporte para la codificación y un potente entorno de integración de utilidades. Este último permite ampliar las funcionalidades de un sistema Drupal de una forma relativamente sencilla (GIL, 2012).

Debido a lo anteriormente mencionado la arquitectura de Drupal es modular. Donde la ampliación de las nuevas funcionalidades se realiza a través de nuevos módulos. Por lo que los algoritmos que se desean incluir deben ser desarrollados en un módulo que se le nombrará “algoritmos”.

Capítulo 2: Características De La Solución Para La Recuperación De Información.

1.5 Arquitectura del módulo “algoritmos”

La arquitectura del módulo “algoritmos” de acuerdo con lo visto en el epígrafe anterior es:

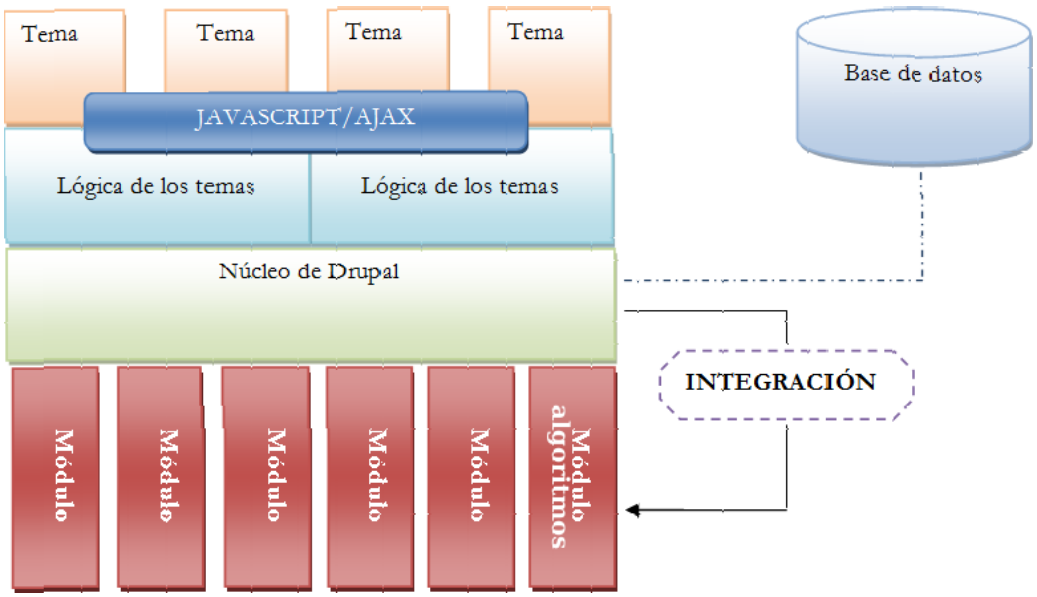

Figura 6: Arquitectura del módulo “algoritmos”

1.6 Modelo de datos

La base de datos para la implementación del módulo “algoritmos” quedaría modificada como se muestra a continuación:

Capítulo 2: Características De La Solución Para La Recuperación De Información.

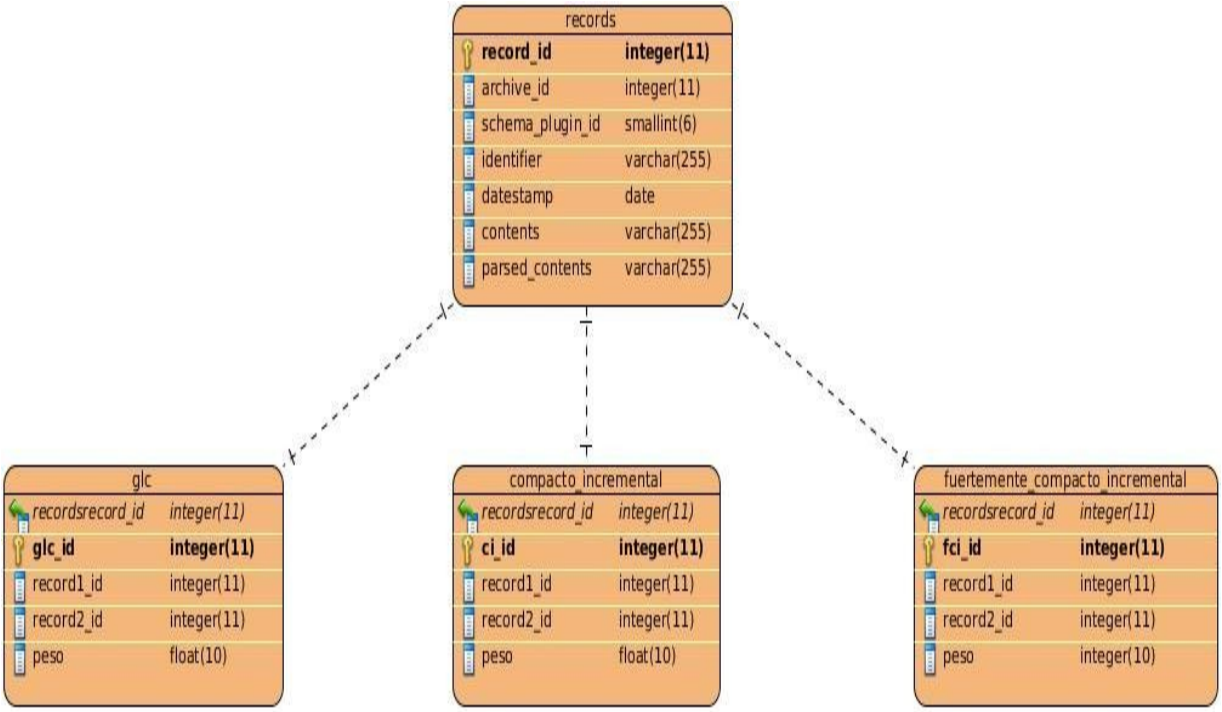

Figura 7: Modelo de datos

Descripción del modelo de datos:

El módulo a desarrollar realiza el agrupamiento de una colección de documentos. En la Figura 7 la colección de documentos con que se trabaja se encuentra en la tabla llama records. Una vez terminada la ejecución de cada algoritmo se guarda en su respectiva tabla de la base de datos su resultado. Para cuando se realice la RI los documentos se encuentren agrupados.

Conclusiones parciales

En el capítulo actual el modelo de dominio facilitó una mejor comprensión de cómo se llevaba a cabo la RI. Además, el modelo de datos brindó la posibilidad de representar la estructura de la base de datos para el funcionamiento de los tres algoritmos.

Capítulo 3: Implementación y prueba.

El presente capítulo contiene cada uno de los aspectos de la implementación y validaciones del módulo a desarrollar teniendo en cuenta los requerimientos definidos en el capítulo anterior.

1.1 Modelo vectorial similitud mediante coseno

El modelo se encuentra representado por la siguiente función (ZAZO, 2002):

$$\text{CosSim}(d_j, d_q) = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$

A continuación se muestra un ejemplo de una colección contenida por tres documentos y se quiere conocer los dos documentos más semejantes:

- D_1: Los algoritmos basados en grafos GLC tienen una complejidad de O(n).
- D_2: Los algoritmos de agrupamientos basados en grafos son una de las técnicas de RI.
- D_3: Una de las técnicas de la RI son los algoritmos de agrupamientos de grafos, los cuales tienen gran complejidad.

Términos->	algoritmos	basados	grafos	GLC	complejidad	O(n)	agrupamientos	técnicas	RI
D_1	1	1	1	1	1	1	0	0	0
D_2	1	1	1	0	0	0	1	1	1
D_3	1	0	1	0	1	0	1	1	1

$\text{CosSim}(d_1, d_2) = ((1*1)+(1*1)+(1*1)+(1*0)+(1*0)+(1*0)+(0*1)+(0*1)+(0*1))/(6*6)$

= 3/36

=0.083

CosSim(d_1, d_3) = ((1*1)+(1*0)+(1*1)+(1*0)+(1*1)+(1*0)+(0*1)+(0*1)+(0*1))/(6*6)

= 3/36

=0.083

CosSim(d_2, d_3) =((1*1)+(1*0)+(1*1)+(0*0)+(0*1)+(0*0)+(1*1)+(1*1)+(1*1))/(6*6)

=5/36

=0.13

Se obtiene como resultado que el D_2 y D_3 son los que más semejanza tienen en la colección.

1.2 Algoritmo GLC

A continuación se observa el pseudocódigo de este algoritmo (PÉREZ, 2008):

Entrada: D= {d_1, d_2, ..., d_n } // colección de documentos

β // umbral de semejanza

Salida: SC // conjunto

SC = ∅; // lista

L = ∅; // lista

para todo $d_i \in$ D hacer

para todo $C_j \in$ SC hacer

si semejante (d_i, C_j , β) entonces L= L ∪ C_j ;

fin

si L = ∅ entonces SC= SC ∪ {d_i};

sino

"Eliminar del SC todos los grupos pertenecientes a L";

C:= {d_i};

para todo C_j ∈ L hacer “Agregar los documentos C_j al grupo C”;

SC= SC ∪ C;

fin

fin

Ejemplo:

D={d_1, d_2,d_3, d_4}

β=0.7

paso 1: para d_i= d_1

L está vacía y SC={d_1}

paso 2: para d_i= d_2

suponiendo que d_1 y d_2 con semejantes entonces

L={d_1}

SC={ } //se eliminan todos los elementos de SC que estén en L

C={d_2}

C={d_1,d_2} //se adicionan los documentos de L a C

SC={d_1,d_2} // queda conformado por la unión de SC y C

Paso 3: para d_i= d_3

C_j= d_1

suponiendo que d_1 y d_3 son semejantes entonces

L= {d_1}

Cj= d_2

suponiendo que d_2 y d_3 no son semejantes no se hace nada y se pasa al siguiente paso

SC = {d_2}

C = {d_1, d_3}

SC = {d_1, d_2, d_3}

1.3 Algoritmo Compacto Incremental

El algoritmo queda representado mediante el siguiente pseudocódigo (PONS, 2004):

Entrada: D= {d_1, d_2, ..., d_n } // colección de documentos

β // umbral de semejanza

Salida: SC // conjuntos compactos

1. Cada vez que se presenta un nuevo objeto O se calcula la semejanza con todos los objetos de los conjuntos compactos existentes.

2. Se seleccionan los objetos que son más β-semejantes a O y aquellos a los que O es el más β-semejante. Si no existen dichos objetos el conjunto {O} es un nuevo conjunto compacto y termina.

3. Los objetos que son β-semejantes a O y los que O es el más β-semejante se analizan:

a) Según sus características van a formar parte de *objetos a unir*, *grupos a procesar* o a *grupos a unir*.

b) Se crea un nuevo conjunto compacto formado por O y por todos los objetos que pertenecen a *grupo a unir*, *objeto a unir* y una parte de *grupos a procesar*.

c) Estos objetos son eliminados de los conjuntos compactos a los que pertenecían.

d) Los conjuntos compactos que quedan vacíos se eliminan.

4. Puede suceder que al eliminar un objeto que pertenece a *grupos a procesar* del conjunto compacto al que pertenecía quede inconexo dicho conjunto y deja de ser un grupo compacto. Para ello se analiza cada objeto que quedo en el conjunto compacto de la siguiente forma:

a) Se selecciona un objeto.

b) Se construye su componente conexa.

c) Se elimina los objetos de esa componente conexa del conjunto compacto.

En este ejemplo se parte del grafo mostrado en la Figura 8, el cual se encuentra formado por 6 conjuntos compactos. Cuando llega el nuevo objeto 18, los conjuntos que construye el algoritmo se muestran en la Figura 9. Como se observa, la llegada del nuevo objeto provoca que se rompan los arcos $<O_0, O_1>$, $<O_0, O_2>$, $<O_{10}, O_{11}>$ y $<O_6, O_7>$, debido a que, ahora, el nuevo objeto es el más β-semejante. Los grupos G_4 y G_5 pertenecen a grupos a unir, pues están conectados con el nuevo objeto y en ellos no se rompió ningún arco. El O_{10} es un objeto a unir pues el nuevo objeto es su más β-semejante y se rompió el único arco que lo conectaba a él. Por último, los grupos G_1 y G_2 pertenecen a grupos a procesar, porque en ellos se rompieron arcos y pueden

perder la propiedad de ser conjuntos compactos.

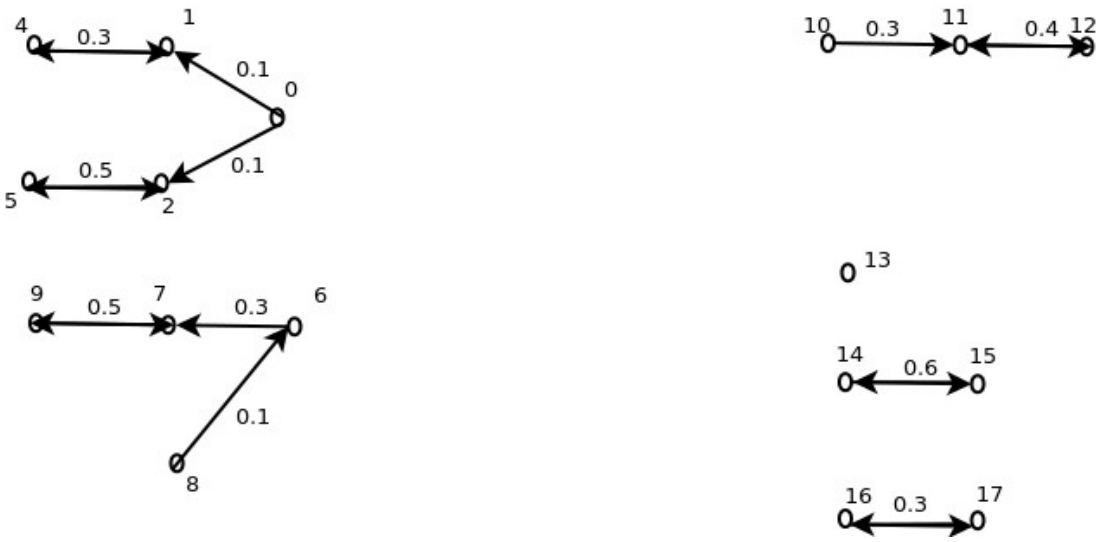

Figura 8: Conjuntos compactos

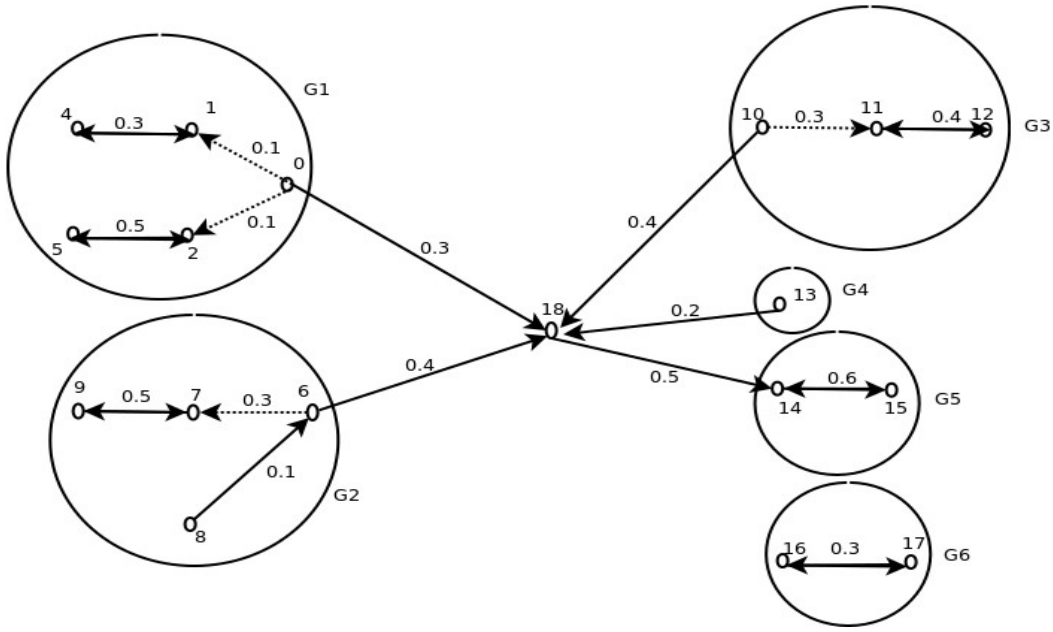

Figura 9: Llegada de un nuevo objeto y sus β-semejantes

Capítulo 3: Implementación Y Prueba.

1.3 Algoritmo Fuertemente Compacto Incremental

El algoritmo queda representado mediante el siguiente pseudocódigo (PONS, 2004):

Entrada: D= {d_1, d_2, ..., d_n } // colección de documentos

β // umbral de semejanza

Salida: SC // conjuntos compactos

1. Cada vez que se presenta un nuevo objeto O se calcula la semejanza con todos los objetos de los conjuntos fuertemente compactos existentes.

2. Se seleccionan los objetos que son más β-semejantes a O y aquellos a los que O es el más β-semejante. Si no existen dichos objetos el conjunto {O} es un nuevo conjunto fuertemente compacto y termina.

3. Los objetos que son β-semejantes a O y los que O es el más β-semejante se analizan:

a) Según sus características van a formar parte de *objetos a unir*, *grupos a procesar* o a *grupos a unir*.

b) Se crea un nuevo conjunto compacto formado por O y por todos los objetos que pertenecen a *grupo a unir*, *objeto a unir* y una parte de *grupos a procesar*. Dicho conjunto va a formar parte de la lista de compactos a procesar.

c) Estos objetos son eliminados de los conjuntos fuertemente compactos a los que pertenecían.

d) Los conjuntos fuertemente compactos que quedan vacíos se eliminan.

4. Para cada objeto que quedó en grupos a procesar:

a) Se construye el conjunto compacto al que él pertenece y se coloca en la lista de compactos a procesar.

b) Se elimina los objetos de esa componente conexa del conjunto fuertemente compacto.

5. Para cada conjunto compacto de la lista de compactos a procesar se construye su cubrimiento en conjuntos fuertemente compactos:

a) Se hallan las componentes fuertemente conexas del grafo de máxima semejanza G_{max} asociado al conjunto, G_c.

b) Se construye el grafo reducido de G_c.

c) Se determina la base B del grafo reducido de G_c, que se encuentra formada por todos los vértices del grafo reducido cuyo grado interior sea nulo.

d) Para cada elemento b_i de B:

i. Se construye el conjunto fuertemente compacto F a partir de b_i, donde en F pertenecen todos los vértices de G_c cuyos vértices correspondientes en el grafo reducido sean descendientes de b_i.

ii. Agregar F al conjunto de conjuntos fuertemente compactos.

1.4 Diagrama de componentes

Los diagramas de componentes muestran como el sistema se encuentra dividido en componentes y la relación que existe entre ellos. Además, ayudan a los desarrolladores a visualizar el camino de la implementación (RIVERA, 2013). A continuación se muestra el diagrama de componentes del módulo desarrollado:

Capítulo 3: Implementación Y Prueba.

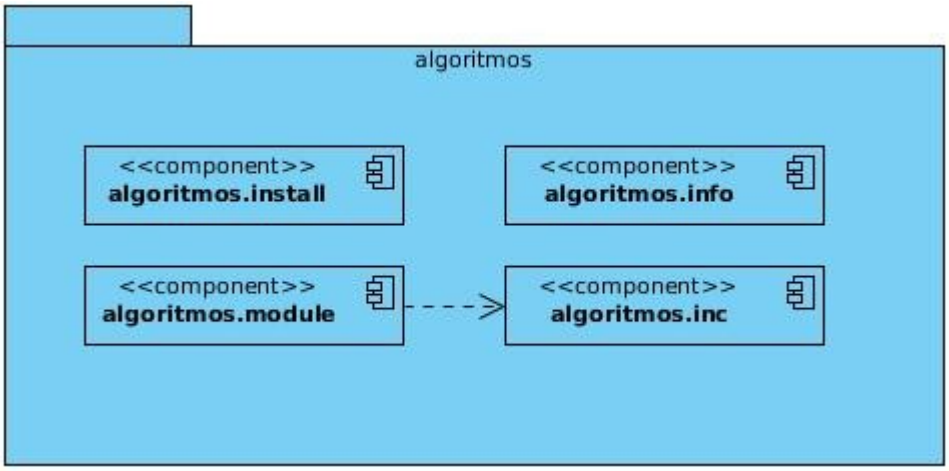

Figura 10: Diagrama de componentes

En la Figura 10 se muestra el diagrama de componentes por el que está estructurado el módulo desarrollado. Donde en el *.module* se crean las tres opciones diferentes de agrupamiento y en *.inc* es donde se implementan estos algoritmos.

1.5 Pruebas

Las medidas de evaluación se realizan mediante la precisión y la exhaustividad. Esta última define la proporción de los grupos que son relevantes que han sido agrupados y permite evaluar la habilidad del módulo para encontrar todos los conjuntos de documentos relevantes de una colección. La pregunta es ¿Son todos los grupos como deben de estar agrupados? Mientras que la precisión define la proporción de los conjuntos agrupados que son relevantes y permite evaluar la habilidad del módulo para agrupar en los grupos la mayoría de los documentos relevantes. La pregunta es ¿Son todos relevantes en el grupo o se filtraron algunos?

Para diferentes muestras de documentos se calcula la precisión y la exhaustividad en los tres algoritmos. Conociendo de cada documento el título, la descripción y las palabras claves.

Ejemplo:

Documento 1:

Título: Modelación y visualización de superficies de terrenos en tres dimensiones.

Descripción: En el presente documento se exponen nuevas técnicas de modelación y visualización interactiva de superficies de terrenos en entornos tridimensionales, con el propósito de modelar superficies que por su extensión puedan ser almacenadas en su totalidad en memoria RAM. Las técnicas clásicas involucran tanto estructuras de datos espaciales como algoritmos para almacenar y manipular modelos poligonales construidos a partir de la información proporcionada por los datos, modelos que constituyen la base para la extracción de triangulaciones multiresolución, con el objetivo de ser visualizadas con la calidad e inmediatez requeridas.

Palabras claves: Inteligencia artificial, modelación, gráficos, topografía, geometría, análisis de datos, algoritmos, técnicas, 3D, geometría del espacio y almacenamiento de datos.

Ejemplo 1: con un umbral de semejanza β igual a 0.07

Prueba del algoritmo GLC:

GLC devuelve la siguiente matriz de adyacencia formando 6 grupos de 7 documentos. En las casillas que se encuentra el número 1 significa que existe semejanza entre los documentos. Donde los documentos 4 y 8 forman un grupo compacto y ambos documentos no son semejantes.

semejantes		1	4	5	6	7	8	10
documento	1							
documento	4						1	
documento	5							

documento	6							
documento	7							
documento	8		1					
documento	10							

Precisión = cantidad de grupos relevantes obtenidos / cantidad de grupos

P= 5/6 = 0.83

Exhaustividad = cantidad de grupos relevantes obtenidos/ cantidad de grupos relevantes

E= 5/7 = 0.71

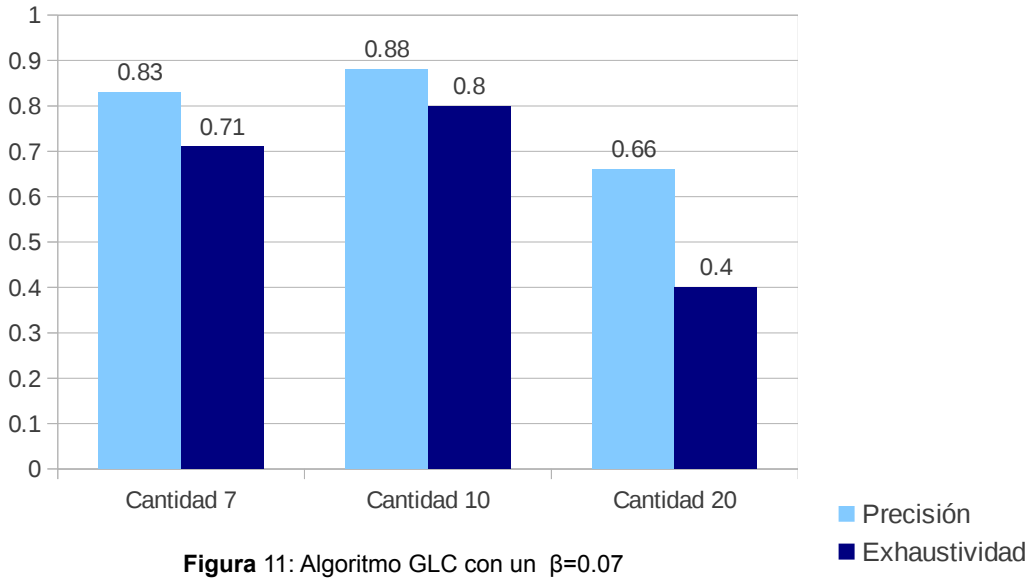

Figura 11: Algoritmo GLC con un β=0.07

Prueba del algoritmo Compacto incremental:

En las casillas que se encuentra un número significa que existe ese número de semejanza entre los documentos. Este algoritmo crea 6 grupos en vez de 7, pues 4 y 8 no son semejantes y se encuentran en un mismo grupo.

semejantes		1	4	5	6	7	8	10
documento	1							
documento	4						0,08	
documento	5							
documento	6							
documento	7							
documento	8		0,08					
documento	10							

P= 5/6 = 0.83

E= 5/7 = 0.71

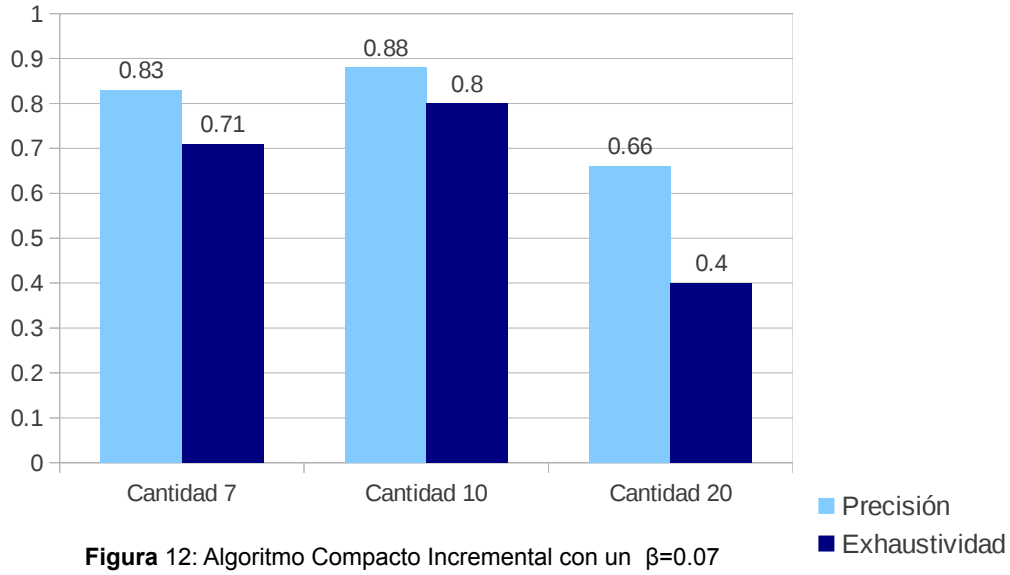

Figura 12: Algoritmo Compacto Incremental con un β=0.07

Prueba del algoritmo Fuertemente Compacto incremental:

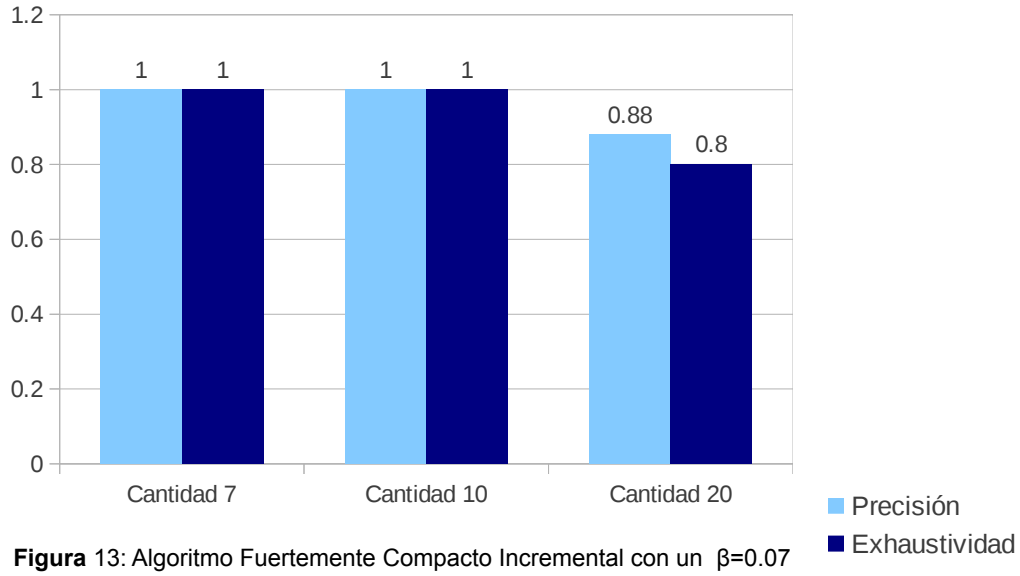

Figura 13: Algoritmo Fuertemente Compacto Incremental con un β=0.07

Ejemplo 2: con un umbral de semejanza β igual a 0.1

Algoritmo GLC

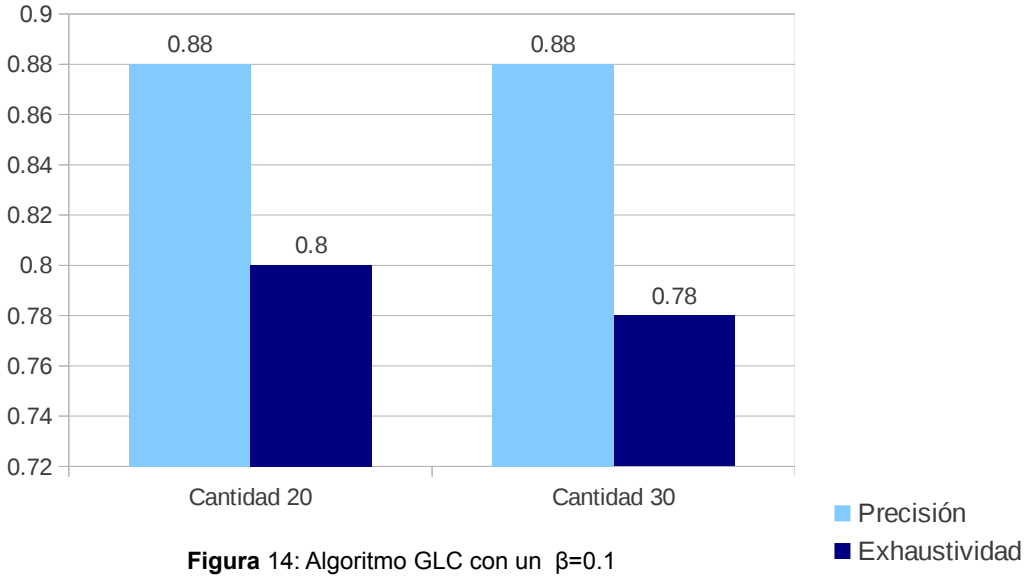

Figura 14: Algoritmo GLC con un β=0.1

Capítulo 3: Implementación Y Prueba.

Algoritmo Compacto Incremental

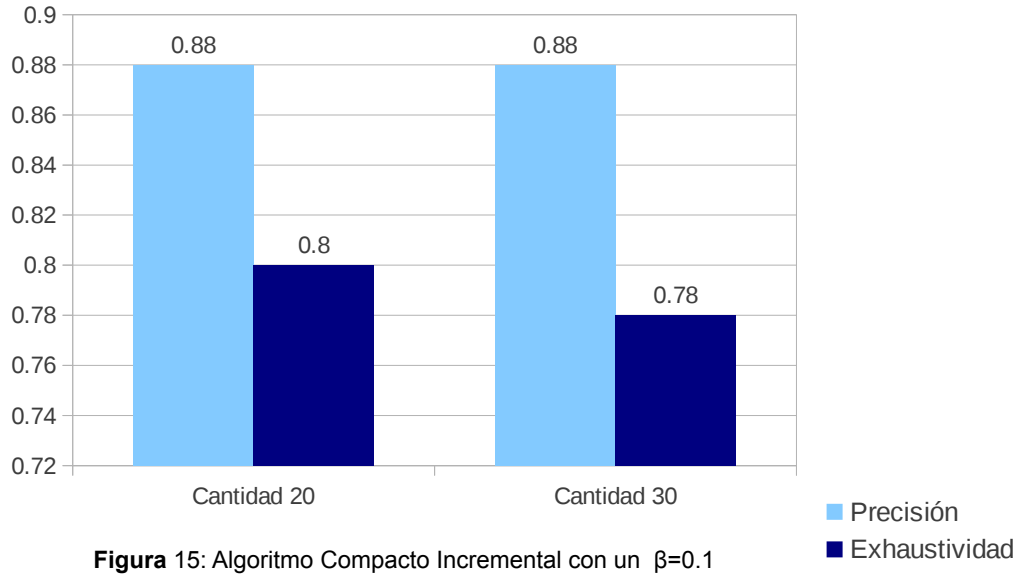

Figura 15: Algoritmo Compacto Incremental con un β=0.1

Algoritmo Fuertemente Compacto Incremental

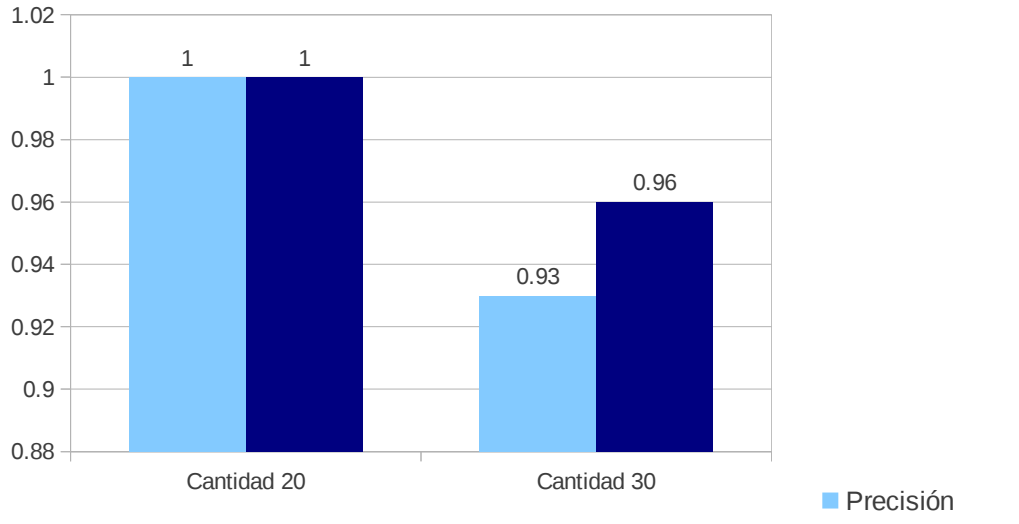

Figura 16: Algoritmo Fuertemente Compacto Incremental con un β=0.1

Capítulo 3: Implementación Y Prueba.

Conclusiones parciales

Cuanto más grande es el número de documentos a agrupar debería aumentar proporcionalmente el umbral de semejanza, pues los algoritmos de esta forma crean grupos más relevantes. Además, la habilidad de agrupar en los algoritmos GLC, Compacto Incremental y Fuertemente Compacto Incremental es aceptable, debido a que en los resultados de los algoritmos se evidencia un número considerable de conjuntos relevantes. Por otra parte, el algoritmo que devuelve los grupos más relevantes es el Fuertemente Compacto Incremental.

Conclusiones

Con el desarrollo del módulo se concluye que:

- Los aspectos teóricos relacionados con las técnicas de agrupamiento para la RI permitieron conocer los autores principales de la RI y realizar el estudio de homólogos.
- El módulo desarrollado implementa los algoritmos GLC, Compacto Incremental y Fuertemente Compacto Incremental.
- Las pruebas demostraron que a medida que aumenta la cantidad de documentos debería aumentar proporcionalmente el valor del umbral de semejanza.

Recomendaciones

Una vez terminado el trabajo de investigación se recomienda:

- Optimizar la implementación de los métodos que se desarrollaron para llevar a cabo los tres algoritmos.
- Tener en un listado la semejanza entre los documentos para cuando se necesite trabajar con las semejanzas no tener que esperar por el tiempo que consume en calcularlos.

Referencias bibliográfica

CASTILLO SEQUERA, José Luis. *Nueva propuesta evolutiva para el agrupamiento de documentos en el sistema de Recuperación de Información.* Tesis doctoral, Escuela Técnica Superior de ingeniería informática, 2010.

BÄR, Marlene. *La Recuperación de Información. Bases de datos como fuentes y recursos de información. Tipos de bases de datos. Localización de las bases de datos* [Página Web]. [Consultado el: 20 de octubre 2012]. Disponible en: http://www.slideshare.net/barmarlene/marlene-tema-1-recuperacin-de-informacin.

MESTRE YENES, Pedro. *Diccionario de la gestión del conocimiento e informática*, 1ra de, Vol 1, Madrid, Informática y computación, 2000, ISBN 84-607-0051-8.

GIL GARCÍA, Reynaldo José*. Algoritmos de agrupamiento sobre grafos y su paralelización.* Tesis doctoral, Universidad de Jaume I, 2005.

PÉREZ SUÁREZ, Ariel; GARCÍA DELGADO, Gail; MEDINA PAGOLA, José E.; MARTÍNEZ TRINIDAD, José Fco y CARRASCO OCHOA, Jesús A. *Algoritmos de agrupamiento para colecciones de documentos*. Ciudad de La Habana, Cuba: 2008.

MOOERS, Calvin N. *Zatocoding applied to mechanical organization of knowledge.* American Documentation, 1951, Vol.2, pág. 20-32. [Consultado el: 30 de enero de 2013]. Disponible en: http://recuperaciond.blogspot.com/2011/09/concepto-de-recuperacion-de-informacion.html.

ZAZO RODRÍGUEZ, Ángel F.; FIGUEROLA PANIAGUA, Carlos G.; ALONZO BERROCAL, José Luis y GÓMEZ DÍAS, Raquel. *Recuperación de Información utilizando el modelo vectorial.* Informe técnico, 2002.

GIMÉNEZ, María Alejandra. *Información básica sobre NetBeans* [Página Web]. Última actualización: 7 de julio de 2012. [Consultado el: 5 de febrero del 2013]. Disponible en:

Referencias Bibliográfica

http://netbeansaccesible.blog*spot.com/*.

ROMEO, Germán. *Gestores de contenidos web (CMS): características* [Página Web]. [Consultado el: 13 de febrero del 2013]. Disponible en: http://www.seas.es/blog/content/gestores-de-contenidos-web-cms-caracteristicas.

MIR CHÁVEZ, Rosana y VALDIVIA FREYRE, Ricardo. *Plataforma de Publicación Web para el periódico Granma.* Tesis de grado, Universidad de las Ciencias Informáticas, 2012.

MARTÍNEZ MORALES, María de los Ángeles. *Investigación sobre la historia de las herramientas CASE* [Página Web]. Última actualización: 12 de septiembre de 2012. [Consultado el: 13 de febrero de 2013]. Disponible en: http://www.slideshare.net/xinithazangels/herramientas-case-14269067.

MORALES, Jeremias. *Lenguajes de programación del lado del cliente* [Página Web]. Última actualización: 20 de octubre del 2011. [Consultado el: 5 de febrero del 2013]. Disponible en: http://www.slideshare.net/JeremiasMorales/22-lenguajes-del-lado-cliente-9782560.

CAMACHO, J. C. *Curso de programación web con HTML* [Página Web]. [Consultado el: 5 de febrero de 2013]. Disponible en: http://es.scribd.com/doc/47778739/114/Caracteristicas-y-ventajas-de-las-CSS.

PACHECO VELASCO, José Evaristo. *Programación web* [Página Web]. Última actualización: 7 de junio 2012. [Consultado el: 5 de febrero 2013]. Disponible en: http://www.prograweb.com.mx/pweb/index.html.

GIL RODRÍGUEZ, Fran. *Experto en Drupal 7 Nivel Avanzado*. Editado por: Forcontu S. L.2012. ISBN-13 (Edición impresa): 978-84-939410-2-4.

KAKES, A. y CASAS, O. *Teoría de grafos.* Editorial Pueblo y Educación, 1987.

MARRERO REYES, Javier. *Solución para la gestión de incidencias en el Organismo Central del Ministerio de la Informática y las Comunicaciones.* Tesis de grado, Universidad de las Ciencias

Referencias Bibliográfica

Informáticas, 2012.

RIVERA ALVA, Eduardo. *Arquitectura de software II. Diagrama de componentes y despliegue* [Página Web]. [Consultado el: 20 de mayo 2013]. Disponible en: http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue.

PONS PORRATA, Aurora. *Desarrollo de algoritmos para la estructuración dinámica de información y su aplicación a la detección de sucesos.* Tesis doctoral, Universidad Jaume I, 2004.

TORRES ALMIRA, Liuba. *Intranet del Ministerio de la Informática y las Comunicaciones*. Tesis de grado, Universidad de las Ciencias Informáticas, 2012.

TOLOSA, Grabiel H. y BORDIGNON Fernando R. *Introducción a la Recuperación de Información. Conceptos, modelos y algoritmos básicos.* Universidad Nacional de Luján, Argentina: 2007.

GIL GARCÍA, Reynaldo José; BADÍA CONTELLES, José Manuel y PONS PORROTA, Aurora. *Algoritmo paralelo para detectar sucesos en noticias en líneas.* En XIV Jornadas de paralelismo. Leganés: septiembre 2003.

Bibliográfica consultada

MESTRE YENES, Pedro, Diccionario de la gestión del conocimiento e informática, 1ra de, Vol 1, Madrid, Informática y computación, 2000, ISBN 84-607-0051-8.

GIL RODRÍGUEZ, Fran. Experto en Drupal 7 Nivel Avanzado. Editado por: Forcontu S. L. 2012. ISBN-13: 978-84-939410-2-4.

PÉREZ SUÁREZ, Ariel; GARCÍA DELGADO, Gail; MEDINA PAGOLA, José E.; MARTÍNEZ TRINIDAD, José Fco y CARRASCO OCHOA, Jesús A. Algoritmos de agrupamiento para colecciones de documentos. Ciudad de La Habana, Cuba: 2008.

PONS PORRATA, Aurora. Desarrollo de algoritmos para la estructuración dinámica de información y su aplicación a la detección de sucesos. Tesis doctoral, Universidad Jaume I, 2004.

GIL GARCÍA, Reynaldo José; BADÍA CONTELLES, José Manuel y PONS PORRATA, Aurora. *Algoritmo paralelo para detectar sucesos en noticias en líneas.* En XIV Jornadas de paralelismo. Leganés: septiembre 2003.

TOLOSA, Grabiel H. y BORDIGNON Fernando R. *Introducción a la Recuperación de Información. Conceptos, modelos y algoritmos básicos.* Universidad Nacional de Luján, Argentina: 2007.

Glosario de términos

Recuperación de Información: es un conjunto de operaciones, que recupera la información contenida en una colección de datos asociada a los criterios de búsqueda definidos por el usuario mediante una consulta.

Algoritmo de agrupamiento: tiene como propósito crear grupos de objetos de manera tal que los objetos pertenecientes a un grupo tengan entre ellos máxima semejanza y con respecto a los objetos pertenecientes a otros grupos, tengan entre ellos menor semejanza.

Algoritmo de agrupamiento basado en grafo: obtienen como resultado final un grafo G (V, E, w) que depende de la semejanza entre los objetos. Dicho grafo se encuentra constituido por los vértices (V) que representan un conjunto de documentos de una colección y el peso (w) de cada arista (E) se encuentra representado por la semejanza entre un par de documentos.

Indización: Es la representación y descripción de la información.

Anexo

Documento 1:

Título: Modelación y visualización de superficies de terrenos en tres dimensiones.

Descripción: En el presente documento se exponen nuevas técnicas de modelación y visualización interactiva de superficies de terrenos en entornos tridimensionales, con el propósito de modelar superficies que por su extensión puedan ser almacenadas en su totalidad en memoria RAM. Las técnicas clásicas involucran tanto estructuras de datos espaciales como algoritmos para almacenar y manipular modelos poligonales construidos a partir de la información proporcionada por los datos, modelos que constituyen la base para la extracción de triangulaciones multiresolución, con el objetivo de ser visualizadas con la calidad e inmediatez requeridas.

Palabras claves: Inteligencia artificial, modelación, gráficos, topografía, geometría, análisis de datos, algoritmos, técnicas, 3D, geometría del espacio y almacenamiento de datos.

Documento 4:

Título: Propuesta del sistema de evaluación del aprendizaje autónomo del idioma inglés en un entorno virtual de aprendizaje en la Universidad de las Ciencias Informáticas.

Descripción: Se tiene en cuenta el enfoque dialéctico-materialista como método general del conocimiento en estrecha relación con la evolución, relación y contradicciones de conceptos tales como autonomía y autoaprendizaje, así como de los procesos cognitivos que tienen lugar en un Centro de Aprendizaje y Servicios de Idiomas Extranjeros. Se emplean como métodos teóricos: el histórico-lógico, el sistémico y el hipotético-educativo. El sistema está desarrollado en EVA Moodle y validado por profesores, usuarios y especialista a través de encuestas.

Palabras claves: Desarrollo de *software*, plataformas virtuales, centros virtuales de recursos,

autoaprendizaje, inglés, sistemas de gestión de aprendizaje, evaluación, Moodle, maestría, proyectos educativos, procesos cognitivos, autonomía, idioma extranjero, aprendizaje virtual y sistemas de evaluación.

Documento 5:

Título: Estrategia de integración para el proyecto de transformación del sistema de identificación, migración y control de extranjeros de la república bolivariana de Venezuela.

Descripción: Desarrollo de una estrategia para la gestión del proyecto de transformación del sistema de identificación, migración y control de extranjeros de la república bolivariana de Venezuela, que favorece una mayor probabilidad en el éxito de su ejecución. Se realiza un estudio de las tendencias que existen para la gestión de proyectos, para que se logre una propuesta que se sustente en fundamentos teóricos de actualidad.

Palabras claves: Informática, gestión de proyectos, estrategia, gestión empresarial, sistemas de detección, Venezuela, proyectos informáticos, integración y proyecto identidad.

Documento 6:

Título: Solución informática para los policlínicos.

Descripción: Se presenta la Gestión de la Integración en la dirección del proyecto para la definición e implantación de una Solución Informática (*Software*, equipamiento tecnológico, redes y comunicaciones, documentación, recursos humanos, mobiliario y conocimientos), que garantiza la solución del problema científico consistente en que el nivel de informatización actual en el policlínico no incluye la gestión de servicios de salud a la población, ni la actualización a nivel comunitario de los registros del sistema de información para la salud (SISalud).

Palabras claves: Informática, salud, sistemas de información, gestión de proyectos, metodología, gestión de la integración, tecnologías de la información y gestión de servicios.

Documento 7:

Anexo

Título: Infraestructura de *software* para el almacenamiento y consulta de la historia clínica electrónica del sistema.

Descripción: Desarrollo de una infraestructura de *software*, basada en estándares internacionales, que facilite el almacenamiento y consulta de la historia clínica electrónica del sistema. Se expone la implementación de la solución y se define la guía de implementación de CD. Se realiza una descripción de la arquitectura, del modelo de datos y del modelo de implementación del visor de historias clínicas electrónicas.

Palabras claves: Informática, sistemas automatizados, infraestructura, modelación, almacenamiento, historias clínicas electrónicas y *software*.

Documento 8:

Título: Control del tránsito en un simulador de conducción de auto.

Descripción: Se da una panorámica de contenidos abordados en trabajos previos y los resultados obtenidos. Se desarrolla un simulador de conducción de autos confeccionado con un enfoque instructivo de entrenamiento y con el realismo necesario para controlar el tránsito en un entorno virtual, haciendo referencia entre la detección de las infracciones del conductor y el movimiento autónomo de los autos.

Palabras claves: Informática, inteligencia artificial, simuladores, informática aplicada y automovilismo

Documento 10:

Título: BlueEye: una plataforma para la gestión de servicios de valor agregado para dispositivos móviles.

Descripción: La plataforma BlueEye es un *software* que se desarrolla con tecnología Java y que gestiona servicios de valor agregado basados en mensajes de texto. BlueEye da la posibilidad de comunicación con uno o varios operadores móviles por medio de distintos

protocolos al mismo tiempo, y brinda un SDK que facilita y agiliza el desarrollo de los nuevos servicios. En el presente trabajo se exponen además las primeras aplicaciones implementadas sobre la plataforma BlueEye, así como las diversas pruebas a que se somete, incluyendo su despliegue en fase de pruebas en el operador móvil Cubacel. Se utiliza la plataforma Java Enterprise Edition y en los diferentes estándares que define Java Community Process e implementa como parte de Java Enterprise Edition, para la realización de diferentes tareas como gestión de mensajes para móviles e implementación de componentes distribuidos.

Palabras claves: Informática, desarrollo de *software*, telefonía digital, control de la calidad, arquitectura de *software*, pruebas, mensajes de textos, dispositivos móviles, gestión de servicios e informática aplicada.

Documento 11:

Título: Servicio de camino mínimo sobre un Sistema de Información Geográfica basado en *software* libre.

Descripción: Desarrollo de un Sistema de Información Geográfica que brinda las funcionalidades básicas (acercar, alejar, información de un punto) y desarrollo de un módulo que permite hacer una búsqueda del camino más corto entre dos direcciones postales, y mostrar el mismo sobre el mapa, solo necesita como entrada que este en formato *shape*. Se especifica una posible forma de manipular mapas muy grandes, obteniendo una representación compacta de un grafo con gran cantidad de vértices y aristas (varios millones), lo que facilita el trabajo con el mapa.

Palabras claves: Aplicaciones web, estructura de datos, algoritmos, geografía, sistemas de información geográfica, posicionamiento geográfico, mapas web, direcciones postales, búsqueda de caminos y algoritmos de reducción.

Documento 12:

Título: Algoritmo para la generación automática de resúmenes de un documento HTML.

Anexo

Descripción: Desarrollo de un algoritmo para la Generación Automática de Resúmenes de páginas web, que utiliza información de marcado presente en el código HTML. Se define una función para determinar la relevancia de un término en el contenido de un documento y se propuso un método para identificar el idioma. Para la evaluación de la calidad del algoritmo se aplican las métricas ROUGE-1, ROUGE-2, ROUGE-L y ROUGE-W y se comparan los resultados con los sistemas comerciales *Copernic Summarizer*, *Pertinence Summarizer* y *Swensun*, donde se obtienen resultados superiores en la métrica ROUGE-1, solo superado por el sistema *Copernic Summarizer* para el resto de las métricas.

Palabras claves: Informática, programación, algoritmos, documentos, HTML, resúmenes e informática aplicada.

Documento 13:

Título: Propuesta de subprocesos de medición basados en CMMI para la Universidad de las Ciencias Informáticas.

Descripción: Esta tesis propone los subprocesos necesarios para el establecimiento de los elementos conductores de un proceso de medición para el área productiva de la Universidad de las Ciencias Informáticas, alineados con los Objetivos de Negocio de la producción. Se describe la vía de definición de los objetivos de medición, indicadores, medidas a recopilar y sus respectivos procedimientos de recolección, almacenaje y verificación, basándose en Medición y Análisis de CMMI y con la utilización de enfoques integrados y recomendaciones a nivel internacional.

Palabras claves: Informática, ingeniería de *software*, gestión de proyectos, gestión de procesos, mediciones, evaluación, CMMI, UCI y métrica de *software*.