UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

CLASIFICACIÓN AUTOMÁTICA DE DOCUMENTOS Y RECUPERACIÓN DE INFORMACIÓN EN COLECCIONES PERSONALES

T E S I S :

PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

METZTLI IBAÑEZ RIVERA

DIRECTOR DE TESIS :

M.C. MA. AUXILIO MEDINA NIETO

HUAJUAPAN DE LEÓN, OAX., DICIEMBRE DEL 2005.

Tesis presentada el 02 de diciembre del 2005

Sinodales:

M. C. Mario Alberto Moreno Rocha

M. C. Irma G. Solís Genesta

M. C. Ricardo Ruiz Rodríguez

Director de tesis:

M.C. Ma. Auxilio Medina nieto

Dedicatoria

Dedico este trabajo a mi familia, gracias por su apoyo incondicional, su amor infinito, su comprensión y por esos ánimos inagotables con los que me impulsan a seguir.

Agradecimientos

A Dios

Gracias por permitirme escribir estas líneas y poner en mi camino a todos los seres que intervinieron en mi formación profesional, la cual inició hace 23 años.

A mi familia

Irma, Apolinar, Citlalli y Zazil. A ti mamá, por siempre darme la fuerza y ánimos para concluir todo lo que empecé, por no flaquear nunca, a pesar de que te dolía más a ti que a mí recordarme que hay deberes. A Polo, porque ha sido mi apoyo paterno, por darme los ánimos y las esperanzas para seguir. A Citlalli y Zazil, mis hermanas, por que ellas son mi inspiración simplemente por eso.

A mis abuelos

Josefa e Inocencio, los cuales cuidaron de esta nieta, aun con sus problemas personales adquirieron otro.

A mis tíos y primos

Principalmente a Benjamín, Hortensia y Olga, los cuales han sido un soporte no sólo familiar, sino también sentimental y económico, gracias tiítos. De igual manera a mis primos Alejandro, Javier, Vicky y Fanny.

A mis amigos de universidad

Agradezco especialmente a los que cursaron algún semestre conmigo, compañeros de desvelos, presiones y todo aquello que implica la vida de estudiante, en especial a Alfredo Reyes Luna y Ramón Maldonado Basilio, aun cuando este último fue mi profesor. Hago énfasis en dos compañeros y amigos: Roberto C. Ríos Medina por haberme brindado su amistad, confianza y apoyo sobre todo al inicio de esta etapa de vida. Francisco J. Hernández Reyes gracias por haberme acompañado en los mejores y peores momentos de mi formación profesional.

A mis sinodales

Muchas gracias a mis sinodales Irma G. Solís Genesta, Mario A. Moreno Rocha y Ricardo Ruiz Rodríguez, por ayudar a que este trabajo estuviera mejor estructurado.

A mi asesora

María Auxilio Medina Nieto, gracias por enseñarme a ser mejor estudiante guiándome en la elaboración de este trabajo, gracias por tu apoyo.

Agradecimientos especiales

Cristino, gracias por escucharme, darme los consejos oportunos, por hacerme un espacio en tu agenda aun con tantas ocupaciones y por la ayuda incondicional que me brindaste.

Finalmente, ofrezco disculpas a todos aquellos que merecieron estar aquí y no los nombro no porque no sean importantes, sino porque se requeriría escribir otro libro mencionando a todos los que ayudaron de alguna manera u otra a que este trabajo sea una realidad.

Índice

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

Los servicios de búsqueda en web facilitan el acceso a múltiples fuentes de información y permiten que los usuarios construyan colecciones personales de documentos digitales. Generalmente estas colecciones están formadas de imágenes, videos o textos de diferente tipo con temas y formatos distintos. Debido a la gran cantidad de documentos, manualmente es difícil encontrar de manera precisa, rápida y sencilla un documento en particular o un grupo de documentos que satisfagan algún requerimiento de información. Los sistemas de recuperación de información tienen como objetivo apoyar esta tarea. En general, estos sistemas se encargan de seleccionar y obtener un conjunto de documentos relevantes.

En la actualidad, la recuperación en colecciones extensas de documentos utiliza algoritmos de búsqueda. Algunos recuperan los documentos que contienen los términos buscados por el usuario; otros más sofisticados, consideran diferentes esquemas de relevancia para cada término además de su aparición. En ambos casos, el usuario obtiene como respuesta un conjunto de documentos potencialmente relevantes. Los últimos han probado ser más eficientes.

Tradicionalmente, la relevancia de un documento es un valor que intenta reflejar en qué medida la información de éste responde a la consulta del usuario. Ésta depende de los valores de relevancia (pesos) que se asignan a cada uno de los términos que describen a dicho documento. En la práctica, es una medida de similitud entre el documento y la consulta.

Frecuentemente, al interactuar con un sistema de información, se obtiene una enorme lista de documentos ordenada según el valor de relevancia, los cuales no siempre cumplen las expectativas de los usuarios. Para mejorar este proceso se emplean mecanismos de clasificación con los cuales se pretenden reducir los tiempos de respuesta e incrementar la precisión de los resultados. La clasificación agrupa documentos de acuerdo a la similitud entre su contenido, permite que las búsquedas se realicen en primera instancia sobre los grupos que comparten mayor similitud con la consulta y no en todos los documentos.

Ante la necesidad de administrar grandes colecciones de documentos personales para recuperar los relevantes, esta tesis describe el diseño e implementación de CREADOC, un sistema encargado de implementar clasificación y recuperación automática de documentos.

1.1 Definición del problema

En una colección, la búsqueda manual de los documentos relevantes es una tarea no factible debido a que consume tiempo, el cual depende del tamaño de la colección y del algoritmo de recuperación. Normalmente se requieren resultados precisos tan rápido como sea posible. Esta tarea se vuelve más compleja en la medida en que el número de documentos es mayor. Por ello es necesario contar con alguna herramienta que soporte la recuperación eficiente de documentos.

1.2 Hipótesis

Es posible organizar de forma eficiente una colección de documentos personales a través de métodos de clasificación y modelos de recuperación de información.

1.3 Objetivo general

Desarrollar una herramienta que apoye la organización de colecciones personales de documentos y la recuperación de los relevantes.

1.4 Objetivos específicos

- Investigar los modelos de recuperación de información y los métodos de clasificación de documentos
- Diseñar una base de datos documental para organizar una colección personal de documentos
- Implementar un método de clasificación de documentos
- Evaluar el método de clasificación
- Recuperar documentos relevantes
- Incorporar un modelo de recuperación de información a la clasificación de documentos

1.5 Propuesta de solución

Esta tesis propone el empleo de métodos de clasificación de documentos, la incorporación de un modelo de recuperación de información que pueda aplicarse a diferentes clases de documentos, con el objetivo de localizar los relevantes de forma eficiente.

Para ello es necesario acceder al contenido de los documentos, consultar sus datos descriptivos e implementar el modelo de recuperación.

1.6 Alcances

- El sistema se ejecutará en la máquina local y se accederá a él a través del navegador
- El sistema trabajará con documentos en formato texto. No se almacenarán los documentos, sólo se procesará el contenido, por tanto, no se pondrán en riesgo los derechos de autor
- El sistema no visualizará el contenido de los documentos, sólo se mostrará el nombre de éstos
- Las acciones sobre la base de datos documental serán:
    - Altas de documentos
    - Consultas de acuerdo al título, autor y palabras clave
    - Consultas por contenido
    - Clasificación de la base de datos documental
    - Consulta de la clasificación de la base documental
- La base de datos documental estará integrada por una colección arbitraria de documentos en Inglés y Español
- La salida del sistema consistirá en una lista ordenada de documentos de acuerdo a un valor de relevancia de los documentos. Esta lista se mostrará como parte de una página web, y como tal, el usuario podrá almacenarla para consultar información de la colección cuando éste lo requiera a través del navegador

La tesis está organizada como sigue: el Capítulo 2 describe recuperación de información y el Capítulo 3 presenta los principales métodos de clasificación de documentos. El Capítulo 4 trata la fase de análisis y diseño. La implementación del sistema se describe en el Capítulo 5. El capítulo 6 reporta la evaluación de los resultados y el capítulo 7 presenta las conclusiones y el trabajo futuro.

Capítulo 2. Recuperación de información

Los sistemas de recuperación de información tienen como objetivo principal seleccionar documentos que respondan a los requerimientos de información o consultas de los usuarios. Como resultado, generan una lista ordenada de documentos de acuerdo a un valor de relevancia.

En la recuperación de información no es suficiente saber qué documentos contienen una o varias palabras de la consulta, (actividad denominada recuperación de datos), sino encontrar documentos relevantes conforme a un grado de similitud entre la consulta y los documentos [Grossman & Frieder 1998]. En sus inicios el objetivo era indexar y buscar documentos útiles en una colección. Actualmente incluye actividades como modelado, clasificación y categorización de documentos, incorpora tareas lingüísticas, visualización de datos y uso de filtros de información.

La recuperación de información modela las consultas como un conjunto de términos que describen la necesidad de información del usuario, interpreta un documento al extraer información sintáctica y semántica y compara las consultas con los documentos. Según [Strzalkowsky 1994], la aplicación efectiva de un modelo de recuperación de información depende de dos actividades principales: la tarea del usuario y la vista lógica de los documentos. Éstas se describen a continuación.

1. *Tarea del usuario*. El usuario necesita traducir su requerimiento de información en una consulta que pueda ser entendida por el sistema. Esto implica especificar un conjunto de palabras con contenido semántico de la información solicitada
2. *Vista lógica de los documentos*. Los documentos se representan como conjuntos de términos denominados *palabras clave*

No todas las palabras clave son igualmente útiles para describir el contenido de un documento. Por ejemplo, si una palabra aparece en muchos documentos, no permite distinguir un documento de los demás. Sin embargo, si ésta aparece en pocos documentos, contribuye de forma significativa a la descripción del contenido.

Las palabras clave pueden ser extraídas por el sistema o especificadas por un usuario. En el primer caso, los documentos se someten a procesos tales como eliminación de palabras vacías, (artículos, conectores o preposiciones), identificación de sustantivos u obtención de raíces léxicas, (proceso conocido también como *lematización)*. Estos procesos minimizan los costos computacionales al reducir la representación de los documentos.

La relevancia de los términos que describen a los documentos se representa mediante un esquema de asignación de pesos [Baeza & Ribeiro 1999]. Generalmente, los pesos son linealmente independientes, aunque algunas técnicas incorporan relaciones de dependencia.

2.1 Modelo booleano

Uno de los modelos más utilizados en Recuperación de Información es el modelo booleano basado en la teoría de conjuntos y en el álgebra booleana. Las consultas se expresan como secuencias de palabras clave que pueden incluir los conectores “NOT”, “AND” y “OR”. A cada palabra clave de la consulta se le asigna un valor de 1 si ésta aparece en el documento y 0 en caso contrario.

De acuerdo a [Salton & Fox 1983], el modelo provee un marco de trabajo fácil de comprender, compuesto por conjuntos de documentos y operaciones estándares sobre conjuntos. Su ventaja principal es que es un modelo simple de formalizar y fácil de implementar. Sin embargo, sólo busca documentos que cumplan con los criterios de la consulta, por lo que puede recuperar muchos o escasos documentos.

Por otro lado, [Baeza & Ribeiro 1999] identifican otras desventajas tales como:

- Considera igual a documentos que contienen una o múltiples apariciones de los términos buscados
- Está basado en un criterio de decisión binario que no considera grados de similitud entre la consulta y los documentos
- No siempre es fácil trasladar la semántica de la información a una expresión booleana

A pesar de esto, el modelo booleano continua utilizándose. En la siguiente sección se describe un modelo que sobrepasa estas desventajas.

2.2 Modelo vectorial

El marco de trabajo de este modelo son vectores de *t* dimensiones y operaciones algebraicas y lineales sobre éstos. Permite asignar valores no binarios a los pesos de los términos en las consultas y documentos. Los pesos se usan para calcular el grado de similitud entre un documento y una consulta.

Como resultado, el modelo vectorial proporciona una lista ordenada de documentos de acuerdo al grado de similitud. En general, el resultado es un conjunto de documentos más preciso que el recuperado por el modelo booleano.

[Baeza & Riberio 1999] describen formalmente al modelo vectorial como sigue:

- El peso de un término ($w_{i,j}$) es un número positivo en el rango [0,1] que intenta reflejar la relevancia del término i en el documento j
- Un documento $\vec{d}_j$ es un vector:

$$\vec{d}_j = \left(w_{1,j}, w_{2,j}, ..., w_{t,j}\right)$$

  donde *t* es el total de términos que describe a un grupo de documentos, llamados *términos índice*
- Una consulta $\vec{q}$ también es un vector:

$$\vec{q} = \left(w_{1,q}, w_{2,q} ... w_{t,q}\right)$$

A los valores $w_{i,j}$ y $w_{i,q}$ que corresponden a términos índice que no aparecen en el documento o la consulta respectivamente, se les asigna un valor igual a cero.

Para evaluar el grado de similitud entre un documento $\vec{d}_j$ y una consulta $\vec{q}$, se considera el coseno del ángulo entre sus vectores y se calcula de acuerdo a la fórmula siguiente:

$$sim\left(d_j, q\right) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \times \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

donde $|\vec{d}_i|$ y $|\vec{q}|$ son los vectores normalizados.

La normalización de los vectores se emplea para que el tamaño de los documentos no influya en el grado de la similitud. En la práctica, la recuperación hace uso de umbrales de similitud para discriminar los documentos relevantes de los no revelantes.

Existen diferentes formas de asignar pesos a los términos de un documento. En esta tesis se emplean la frecuencia y frecuencia inversa descritas en [Baeza & Riberio 1999], las cuales se definen como sigue.

Sea *N* el número total de documentos, n_i el número de documentos que contienen el término k_i. A partir de la frecuencia simple *($freq_{i,j}$),* (*l*a cual es el número de veces que el término k_i está contenido en el documento d_j), se obtiene el valor de la *frecuencia normalizada ($f_{i,j}$)* como:

$$f_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}}$$

donde $\max_l freq_{l,j}$, es el valor máximo de frecuencia de un término del documento d_j. Si el término k_i no aparece en el documento d_j entonces $f_{i,j}=0$.

La frecuencia inversa del término k_i en el documento j (*idf*) es:

$$idf = \log\left(\frac{N}{n_i}\right)$$

Un esquema común de pesos de términos considera:

$$w_{i,j} = f_{i,j} \times \log\left(\frac{N}{n_i}\right)$$

Este valor se conoce como el factor *TF/IDF*, (de las siglas de *Term Frequency* e *Inverse Document Frequency*).

Las ventajas del modelo son:

- El esquema de pesos de términos mejora la precisión de la recuperación
- Los pesos no binarios permiten recuperar documentos que se aproximan a las condiciones establecidas en la consulta
- El conjunto de documentos recuperados pueden ordenarse de acuerdo al grado de similitud

Teóricamente la desventaja principal del método de espacios vectoriales es que los pesos de los términos se asumen linealmente independientes. Existen otros métodos de recuperación, sin embargo, en la actualidad se considera la mejor alternativa y una fuerte estrategia para clasificar colecciones de documentos [Zazo et. al. 2002]. Esta tesis optó por usar este método para recuperar información a nivel de contenido. La recuperación de metadatos, emplea un formato que ha sido utilizado ampliamente en bibliotecas digitales y en la web. Éste se describe en la siguiente sección.

2.3 Iniciativa de Metadatos Dublin Core

El término *meta* es una palabra de origen griego que tiene múltiples interpretaciones como *junto a*, *después*, *más allá* o *siguiente*. En sistemas de información, metadatos se interpreta como datos sobre otros datos. Este término se emplea también para designar a la información que describe recursos de información. Un recurso puede ser un documento, un enlace, una imagen o cualquier otro elemento el cual pueda identificarse de forma unívoca en la web.

El número y la diversidad de documentos en bibliotecas digitales ocasionan que los sistemas de recuperación produzcan resultados no relevantes. Los estándares de metadatos surgen como una alternativa para tratar de elevar la calidad de los resultados. El estándar de metadatos seleccionado en esta tesis lo propone la iniciativa DCMI, (*Dublin Core Metadata Initiative*).

DCMI es una organización dedicada a la promoción y difusión de normas interoperables sobre metadatos y desarrollo de vocabularios especializados para describir recursos que permitan el uso de sistemas de recuperación de información inteligentes. La misión de DCMI es facilitar la búsqueda de recursos utilizando Internet a través de actividades como las siguientes [URL01]:

- Desarrollar estándares de metadatos para la recuperación de información
- Definir marcos para la interoperabilidad entre metadatos
- Facilitar el desarrollo de metadatos en diversas disciplinas consistentes con estándares y mecanismos de interoperabilidad

A continuación, se listan los elementos de DCMI para describir los recursos en letras cursivas, seguidos de prácticas recomendadas para su uso.

- *Title (Título).* Es el nombre del recurso

- *Creator (Creador).* Entidad responsable del contenido del recurso. El creador de un recurso puede ser una persona, una organización o un servicio

- *Subject (Tema).* Describe el contenido del recurso, se expresa con palabras o frases clave, incluso códigos de clasificación

- *Description (Descripción).* Explicación del contenido del recurso. Puede se un resumen, una tabla de contenidos, una referencia a una representación gráfica o una explicación en lenguaje natural

- *Publisher (Editor).* Entidad responsable de que el recurso esté disponible. Los editores pueden ser una persona, una organización o un servicio

- *Contributor (Colaborador).* Entidad responsable de realizar contribuciones al contenido de un recurso. Un colaborador puede ser una persona, una organización o un servicio

- *Date (Fecha).* Fecha de una circunstancia relativa al ciclo de vida del recurso. Comúnmente, la fecha se asocia con la creación o disponibilidad del recurso

- *Resource Type (Tipo de recurso).* Naturaleza o género del contenido del recurso. El tipo se refiere a términos que describen categoría, funciones, géneros o niveles de agregación del contenido

- *Format (Formato).* Manifestación física o digital de un recurso. Incluye tipos de medios o dimensiones de un recurso. El formato puede usarse para identificar el software, hardware, u otros equipamientos necesarios para visualizar/presentar u operar el recurso

- *Resource Identifier (Identificador).* Referencia inequívoca a un recurso dentro de un contexto dado. Identifica el recurso por medio de una cadena o número adaptado a un sistema formal de identificación

- *Source (Fuente).* Referencia a un recurso del cual deriva el recurso que se describe. El recurso actual puede derivar de una o más fuentes. Se recomienda identificar el recurso referenciado por medio de una cadena o número conforme con un sistema de identificación formal

- *Language (Idioma).* Idioma del contenido intelectual de un recurso. Por ejemplo, se incluye *en* o *eng* para el Inglés y *es* o *spa* para el Español o Castellano

- *Relation (Relación).* Referencia a un recurso relacionado. Se recomienda identificar los recursos relacionados mediante un cadena o número conforme a un sistema de identificación formal

- *Coverage (Cobertura).* La magnitud o el alcance del contenido de un recurso. La cobertura incluye la localización espacial, (nombre de un lugar o coordenadas geográficas), periodo o fecha y jurisdicción, (por ejemplo, denominación de una entidad administrativa)

- *Rights Management (Derechos).* Información sobre los derechos legales que afectan el uso del recurso. Los derechos incluyen una declaración de gestión para el recurso o hacen referencia a un servicio que proporcione dicha información. La información sobre los derechos incluye los derechos de Propiedad Intelectual (IPR*, Intellectual Property Rights*), derechos de autor (copyright), entre otros. Si el recurso no contiene este elemento, se asume que puede utilizarse libremente

Capítulo 3. Clasificación de documentos

La recuperación de información es una tarea lenta en colecciones extensas de documentos, más aún cuando éstos no están organizados. Para acelerar este proceso, se emplean técnicas de clasificación de manera que la recuperación se lleve a cabo sólo en algunos grupos.

En general, existen dos tipos de clasificación: categorización y agrupamiento. A diferencia del segundo, la categorización requiere mayor supervisión humana. Este capítulo describe los principales métodos de agrupamiento y sus características más representativas. En adelante, se asume que los datos a agrupar representan documentos.

De acuerdo a [Fung et al. 2004], las características deseables de un método de agrupamiento son las siguientes:

- *Dimensionalidad elevada*: Capacidad del método para manejar grandes cantidades de documentos o un número elevado de términos descriptivos

- *Escalabilidad*: Esta característica indica que la precisión del método no depende de forma significativa del número de documentos

- *Precisión*: Un buen método de agrupamiento debe tener una similitud alta en el interior de sus grupos y baja en comparación con los grupos del exterior

- *Búsqueda fácil con descripción significativa de grupos, (navegación sencilla)*. Los grupos resultantes deben estar organizados y contar con información descriptiva que facilite la búsqueda de documentos

- *Conocimiento mínimo del dominio.-*. Algunos métodos requieren que el usuario especifique los valores de parámetros de entrada tal como el número de grupos que se desea formar. Sin embargo, el usuario no siempre tiene este conocimiento del dominio

Existen diferentes métodos que pueden aplicarse para agrupar documentos, algunos se muestran en la Figura 1. Esta sección describe brevemente sus características principales.

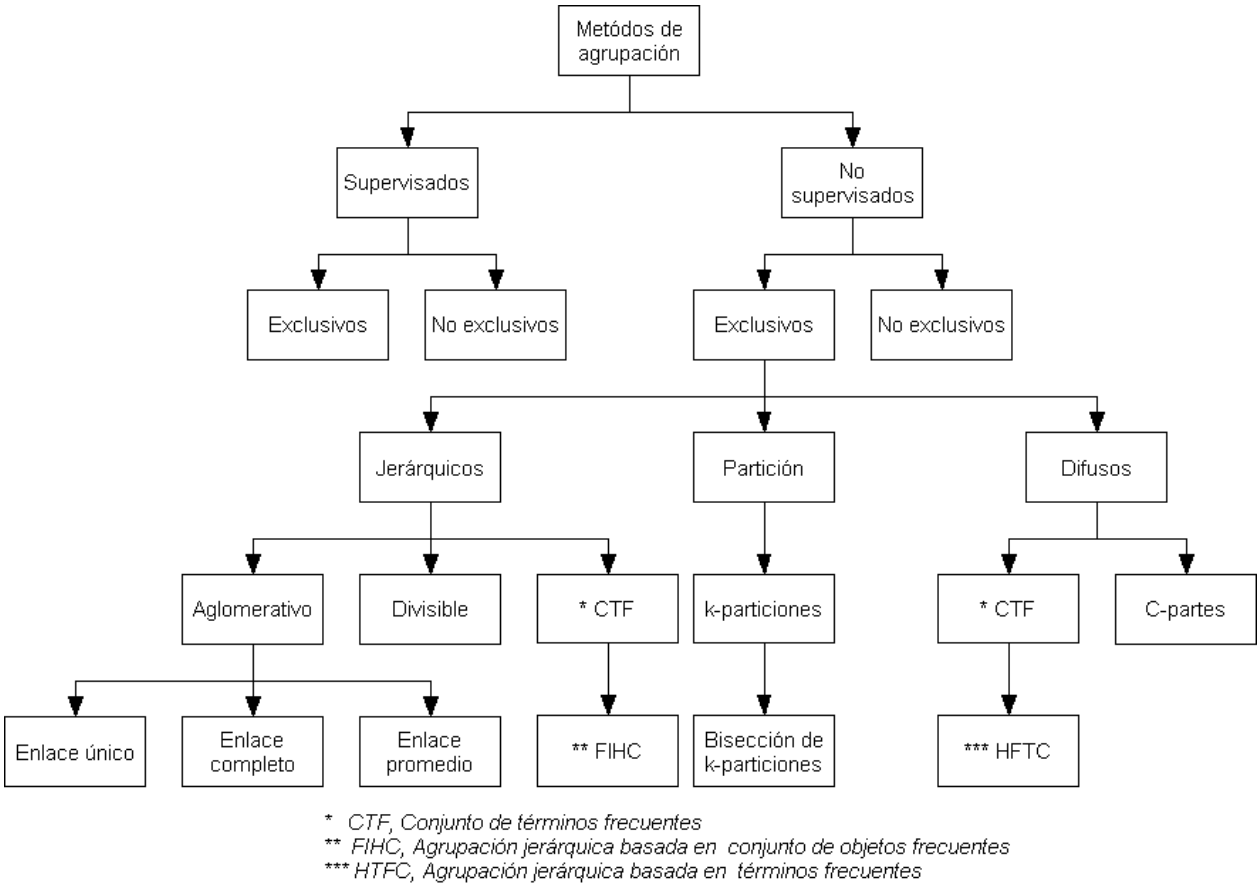


Figura 3.1. Esquema que muestra la clasificación de los métodos de agrupamiento**.**

Una distinción en los métodos de agrupamiento es la exclusividad, la cual significa que cada objeto pertenece a un solo grupo; otra es la estructura de organización, la cual puede ser jerárquica o de partición. [Brücher et. al. 2002] presentan un análisis detallado de estos métodos.

3.1 Métodos de agrupamiento jerárquicos

La técnica de ordenamiento jerárquico une (aglomera) o divide (particiona) grupos que forman una estructura de árbol. Los grupos son mutuamente excluyentes.

3.1.1 Métodos jerárquicos aglomerados

Cada grupo contiene inicialmente un documento. Une grupos al realizar los pasos siguientes:

1. Encuentra los dos grupos más cercanos y los une en un solo grupo

2. Sustituye los dos elementos del paso anterior por un elemento que describa al grupo. Calcula nuevamente las distancias entre este grupo y los demás
3. Regresa al paso 2 si hay más de un grupo

De acuerdo a [Cimiano et. al. 2004], los métodos aglomerados construyen una jerarquía de abajo hacia arriba. Para seleccionar los grupos que se van a unir, [Jain et al. 1999] definen los métodos siguientes:

a) *Enlace único.-* La distancia entre dos grupos se determina por la similitud de los dos objetos más cercanos, (vecinos más cercanos) en los diferentes grupos. Este método es el más conocido dentro de los métodos jerárquicos

b) *Enlace completo.-* La distancia entre grupos se determina por la menor similitud entre dos objetos en grupos diferentes. El método no es apropiado si el grupo tiende a ser extenso

c) *Enlace promedio.-* La distancia entre dos grupos se calcula como el promedio de las distancias entre todos los puntos en los grupos. Es una combinación del método de enlace único y de enlace completo. Es menos sensible a entradas erróneas, porque si un objeto es completamente distinto de otros, no afecta el agrupamiento

3.1.2 Métodos jerárquicos divisibles

Estos métodos construyen una jerarquía de arriba hacia abajo, parten de un conjunto de documentos y lo dividen sucesivamente hasta llegar a elementos individuales. El método típico de esta clase es el método de Ward, el cual tiene dos versiones: *divisible* y *aglomerado*. La versión divisible realiza un análisis de las diferencias para evaluar las distancias entre grupos. La asociación del grupo se evalúa calculando la similitud promedio, lo cual implica encontrar el término medio de cada grupo y la similitud de los objetos que pertenecen al grupo. El método es eficiente, sin embargo, tiende a crear grupos pequeños, lo cual implica una estructura de árbol extensa que dificulta la búsqueda [Fung et al. 2004]. La versión aglomerada no se usa frecuentemente y por ello no se describe en este documento.

En general, los métodos jerárquicos requieren poco esfuerzo para fusionar y dividir los grupos, sin embargo, no son reversibles, es decir, una vez que se forman los grupos, no hay manera de modificarlos posteriormente. Otra desventaja es que el cálculo de la similitud entre cada par de grupos no es escalable si se consideran grandes cantidades de documentos.

3.2 Métodos de agrupamiento por partición

El concepto básico de partición es la descomposición de un conjunto de objetos en *k-particiones* o grupos sin imponer una estructura jerárquica. Cada grupo se representa por un objeto central llamado *centroide* a partir del cual se ordenan las descripciones de los objetos del grupo. [Fung et al. 2004] analiza las características principales de estos métodos, las cuales se citan a continuación.

a) *k*- particiones básico

El método *k*-particiones representa la categoría de métodos de partición de grupos. Este método refina de forma iterativa un conjunto seleccionado de objetos hasta que los miembros del grupo se estabilizan.

El algoritmo sigue los pasos siguientes [Jain et al. 1999]:

1. Se define el número de grupos *k* que se van a formar
2. Se escogen *k* documentos arbitrariamente o por prioridad como centro de los grupos
3. Cada documento se asigna al centro del grupo más cercano usando la métrica de la distancia euclidiana, trata de minimizar la similitud promedio entre los documentos más cercanos maximizando la similitud entre grupos
4. Se recalcula el centro del grupo para formar más grupos tomando en cuenta el nuevo elemento
5. Se ejecuta el paso 3 y 4 hasta que no haya cambios en los centros de los grupos

Este método no es adecuado para hallar grupos de tamaños muy variados.

b) Bisección *k*-particiones

El método de bisección *k*-particiones funciona mejor en términos de precisión y eficiencia que *k*-particiones básico. Este método opera como sigue:

1. Selecciona un grupo a dividir
2. Emplea *k*-particiones básico para crear dos subgrupos
3. Se regresa al paso 1 hasta alcanzar el número de grupos deseados

Ambos métodos son eficientes y escalables, su complejidad es lineal con respecto al número de documentos y se usan ampliamente debido a que son fáciles de implementar. Su desventaja principal es que una estimación incorrecta del número de grupos conducen a una baja precisión.

c) Un solo paso

Inicia con un conjunto de grupos vacíos. Crea una partición en el conjunto de datos de acuerdo a los pasos siguientes [Jain & Dubes 1998]:

1. El primer objeto se considera el centro del grupo. Este objeto se elige al azar o se predefine
2. Para el siguiente objeto, se calcula la similitud con cada centroide del grupo existente usando algún coeficiente de similitud
3. Si el cálculo más elevado es más grande que algún valor específico, se agrega el objeto al grupo correspondiente y se redetermina el centroide; de otro modo, el objeto se usa para iniciar un nuevo grupo
4. Si quedan objetos que aún no se han agrupado regresar al paso 2

Como su nombre lo indica, este método requiere sólo un paso a través del conjunto de datos, es eficiente para un procesamiento serial. Su desventaja es que el agrupamiento depende del orden en el cual se procesan los documentos. El primer grupo formado es mayor que aquellos formados después de ejecutar el método.

d) El vecino más cercano

Es un método iterativo, similar al método jerárquico de enlace único. Usa la distancia del grupo más cercano como un umbral para determinar si se agregan objetos a un grupo existente o se crea un nuevo grupo. Su implementación es sencilla, sin embargo no se aplica a colecciones reducidas de objetos. Es un método lento [Jain et al. 1999].

En general, los métodos de partición son convenientes para conjuntos grandes de objetos para los cuales la construcción de una estructura en forma de árbol tiene un costo elevado. Sin embargo, su precisión depende de la elección del número de grupos deseado.

3.3 Métodos de agrupamiento difuso

El agrupamiento difuso genera un número específico de particiones con límites difusos. No es exclusivo, cada objeto pertenece a uno o más grupos simultáneamente pero en diferente medida, esto es, permite negociar el traslape de fronteras. A continuación se describen algunos métodos difusos [Fung et al. 2004].

a) C-partes

Elige al azar el centro del grupo y mejora la función objetivo de forma iterativa. El método muestra una tendencia a particionar los grupos con un número similar de objetos. Los grupos centrales iniciales influyen en el resultado [Jain et al. 1999].

b) Métodos basados en conjuntos de términos frecuentes

El método HFTC, siglas cuyo significado proviene de *Hierarchical Frequent Term-based Clustering*, es un algoritmo basado en agrupamiento jerárquico. HFTC forma grupos etiquetados con los términos que tienen una mayor frecuencia en el conjunto de documentos, permitiendo que los documentos pertenezcan a diferentes grupos, es decir un documento pueda ser clasificado bajo varios temas diferentes [Florian et al. 2002].

Este método selecciona el mínimo conjunto de objetos frecuentes que representa al grupo, minimiza el traslape de términos en grupos que comparten documentos. El resultado del agrupamiento depende del orden en que se seleccionan los objetos, pueden elegirse de entre varias heurísticas. HFTC es comparable con bisección *k*-partes en términos de precisión, sin embargo, no es escalable.

3.4 Agrupamiento jerárquico basado en conjuntos de términos frecuentes

En los métodos jerárquicos clásicos y en los métodos de partición, la similitud entre dos conjuntos es muy importante para construir un grupo. Se dice que estos métodos están *centrados en documentos*. Un método centrado en grupos es FIHC, siglas de *Frequent Itemset-based Hierarchical Clustering*, dado que las medidas de similitud de grupo usan la frecuencia de los términos de forma que documentos en el mismo grupo comparten más términos comunes que documentos de otros grupos. Es un método escalable [Fung et al. 2003].

FIHC emplea el concepto de conjunto de términos frecuentes (*frequent itemsets*) para referirse a un conjunto de términos que ocurren en una fracción mínima de documentos. El método identifica los términos que pertenecen a varios grupos y con ellos construye grupos iniciales.

Un documento se modela como un vector característico, el cual contiene términos descriptivos y su frecuencia. Por ejemplo, las filas de la Tabla 3.1 forman dos vectores característicos para dos documentos descritos por los términos *recuperación, información, bibliotecas* y *digitales*.

Tabla 3.1. Ejemplo de vectores característicos.

	recuperación	**información**	**bibliotecas**	**digitales**
Documento1	3	2	1	0
Documento2	0	3	4	5

Antes de iniciar el algoritmo, el usuario debe especificar dos parámetros de clasificación: el porcentaje mínimo de aparición de un término en la colección para que sea considerado término frecuente y el porcentaje mínimo de aparición de un conjunto de términos frecuentes. El primero se conoce como *frecuencia mínima global o soporte global* y el segundo *frecuencia mínima grupal o soporte de grupo.*

El método FIHC se resume en tres fases:

1. *Construcción de grupos*

Se identifican todos los términos que aparecen en el conjunto de documentos con un porcentaje mayor o igual a la frecuencia mínima global establecida por el usuario, con ello se obtiene la lista global de términos frecuentes. Para cada término frecuente se construye un grupo que incluye todos los documentos que contienen ese término. Los grupos iniciales comparten documentos dado que un documento puede contener múltiples términos frecuentes.

Los documentos se comparan para encontrar los conjuntos de términos frecuentes comunes que tienen un porcentaje mayor o igual a la frecuencia mínima grupal. La obtención del conjunto de términos frecuentes se basa en el uso del algoritmo *a priori*, el cual se describe en [Agrawal & Ramakrishnan 1994] Posteriormente cada documento se asigna al grupo con el cual tenga mayor similitud y comparta el mayor número de términos frecuentes de forma que cada documento pertenezca a un solo grupo. FIHC identifica cada grupo con una etiqueta formada por el conjunto de términos frecuentes. Todos los documentos que no contienen términos frecuentes o que no comparten la similitud mínima requerida para pertenecer al grupo forman un grupo con etiqueta nula descendiente directo de la raíz.

Sea G_i el grupo *i*, d_j el documento *j*, *x* el conjunto de términos frecuentes globales que pertenecen al grupo y *x*' el conjunto de términos frecuentes globales que no pertenecen al grupo. Para determinar a qué grupo G_i pertenece el documento d_j, se aplica la fórmula siguiente:

$$pertenencia(G_i \leftarrow d_j) = \left[\sum_x n(x) \times soporte\_grupo(x)\right] - \left[\sum_{x'} n(x') \times soporte\_global(x')\right]$$

donde *n(x)* y *n(x')* es la suma de las frecuencias (*TFxIDF*) de *x* y *x'* respectivamente en el vector característico del documento *j*. El grupo con el cual obtenga un valor máximo de pertenencia, es al grupo al que se asigna el documento. En caso de empate, el documento se asigna al grupo cuya etiqueta contiene mayor número de términos.

2. Construcción del árbol de grupos

La construcción del árbol se inicia de abajo hacia arriba, esto es, se busca el mejor padre de cada grupo. Cada grupo (excepto el nodo raíz) tiene exactamente un padre. El tópico de un padre es más general que el tópico de un grupo hijo y comparten cierta similitud. La etiqueta del padre es un subconjunto de las etiquetas de los grupos hijos. El criterio para seleccionar el mejor padre es similar al de escoger el mejor grupo para el documento, es decir se busca el subgrupo con el que comparte mayor similitud.

3. Poda del árbol de grupos

Como resultado de la clasificación se obtiene un árbol de grupos que puede ser ancho o profundo, lo cual dificultará la búsqueda. La poda del árbol consiste en quitar los grupos demasiado específicos para que la búsqueda sea más eficiente. La idea es que si dos grupos hermanos o un grupo padre y su hijo son muy similares, pueden formar un solo grupo.

a) Poda de hijos

FIHC permite la creación de grupos descendientes a varios niveles. A niveles intermedios pueden existir grupos que contengan pocos o ningún documento. La poda en profundidad elimina estos grupos. Para ello obtiene la similitud entre un grupo y el grupo que lo antecede.

Sean G_i y G_j dos grupos, $doc(G_j)$ es la combinación de todos los documentos del subárbol G_j como un único documento. X representa la frecuencia global de los términos en el documento G_j que son objetos frecuentes en G_i y X' representa los términos globales frecuentes en el $doc(G_j)$ que no son términos frecuentes en G_i. La similitud del grupo G_i con el grupo G_j, se calcula mediante:

$$similitud\left(G_i \leftarrow G_j\right) = \frac{pertenencia(G_i \rightarrow doc(G_j))}{\sum_x n(x) + \sum_{x'} n(x')} + 1$$

donde $n(X)$ es el peso de la frecuencia de X en el vector característico de $doc(G_j)$; $n(X')$ es el peso de la frecuencia de X' en el vector característico $doc(G_j)$.

El valor de la similitud es un número en el rango [0, 2]. Si el valor de similitud que comparten ambos grupos es mayor que 1, entonces éstos se unen.

La similitud que comparten ambos grupos se calcula como:

$$Inter\_similitud(G_i \leftrightarrow G_j) = \left[similitud(G_i \leftarrow G_j) * similitud(G_j \leftarrow G_i)\right]^{1/2}$$

b) Poda de hermanos

Esta poda sólo se aplica en el primer nivel de la estructura jerárquica, esto es para minimizar el número de términos que describen a los grupos principales. En este nivel, se emplean las funciones de similitud y similitud entre grupos descritas en la poda de hijos.

Una vez revisados los distintos algoritmos para agrupar documentos, en esta tesis se optó por FIHC, el cual no requiere la supervisión de un experto, (es no supervisado), su eficiencia no depende del conocimiento previo del usuario tal como el número de grupos y produce una estructura jerárquica que facilita la búsqueda de documentos relevantes. Además, es un algoritmo escalable, lo cual implica que su desempeño no depende en gran medida del tamaño de la colección.

3.5 Trabajo relacionado

El sistema CREADOC es un sistema que involucra conceptos de recuperación de información y algoritmos para clasificar documentos de forma automática. A continuación se describen algunos trabajos de investigación y herramientas que emplean métodos similares.

3.5.1 *Vivísimo*

Vivísimo[1] es un meta-buscador en línea desarrollado por la Universidad de Carnegie Mellon que emplea un algoritmo de agrupamiento basado en la similitud textual y lingüística. El algoritmo es propietario. El motor de búsqueda considera únicamente títulos y resúmenes de documentos en web, no se analizan los documentos completos u otros parámetros [URL 01].

El agrupamiento coloca documentos en el mismo grupo dependiendo de la similitud entre ellos. El método complementa su decisión de similitud con la ayuda de supervisión humana llevada a cabo por los programadores. No se utiliza una taxonomía predefinida o control de vocabulario, sólo se usa la descripción del documento. Este método no es exclusivo, de manera que un documento puede pertenecer a varios grupos dentro de una jerarquía.

La salida de este software es una lista ordenada de acuerdo a la similitud con la consulta. *Vivísimo* permite realizar búsquedas sobre toda la web o sobre algunos sitios específicos como noticias, buscadores y portales. En la Figura 3.2 se muestra la pantalla principal de *Vivísimo*, en la cual se teclean los términos que se desean buscar, y en la que opcionalmente puede seleccionarse el dominio de la búsqueda.

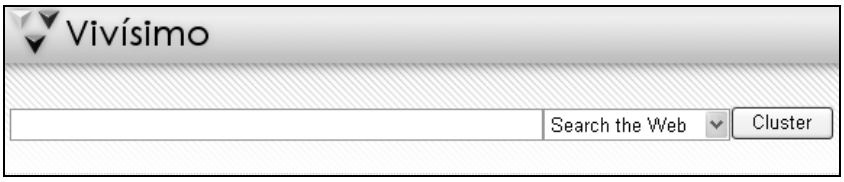


Figura 3.2. Pantalla principal de software *Vivísimo.*

Los resultados de la consulta realizada se muestran en una página dividida en tres secciones como se muestra en la Figura 3.3. En la sección superior, el usuario puede realizar otra consulta. La sección inferior izquierda contiene la lista de los grupos recuperados (nombre y número de documentos contenidos en el grupo) que se relacionan con la consulta realizada. La lista de grupos puede expandirse.

La sección inferior derecha muestra las páginas que tienen mayor similitud con la consulta. Una página puede aparecer en más de un grupo. Cada grupo puede contener subgrupos.

[1] Vivìsimo//Vivisimo Clustering – automatic categorization and meta-search, disponible en: http://www.vivisimo.com

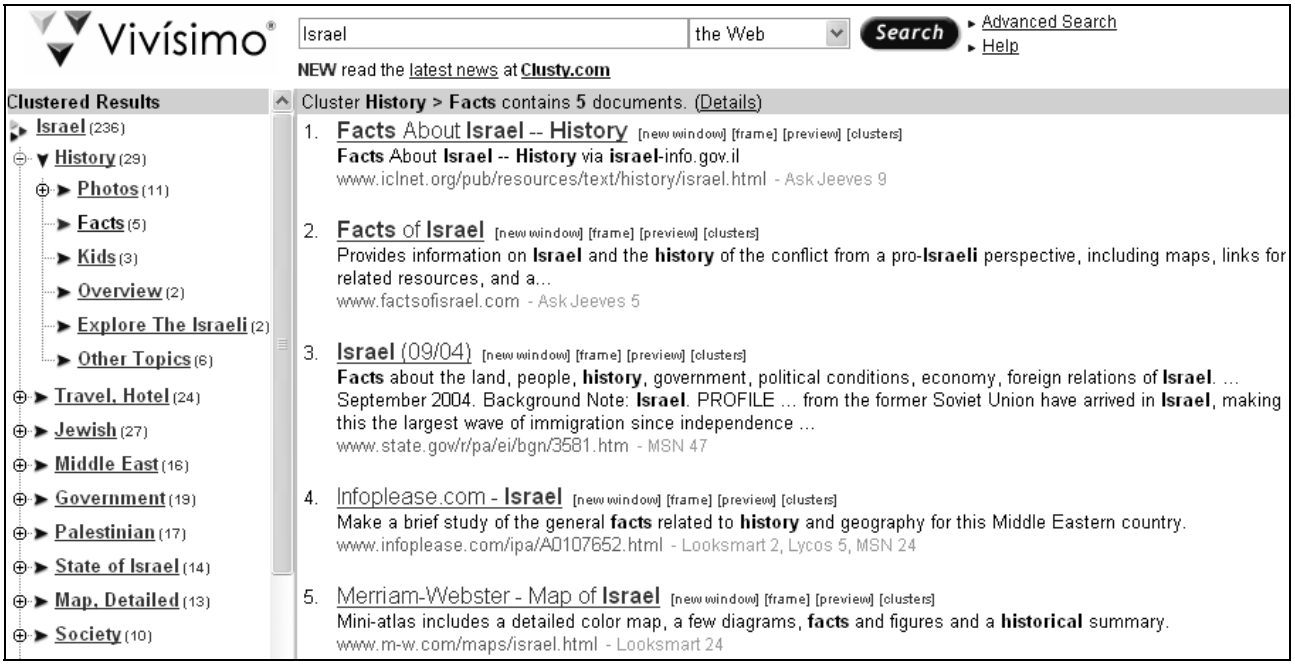


Figura 3.3. Interfaz de resultados de *Vivísimo*.

La Figura 3.3 muestra el resultado de la consulta *Israel* que contiene un total de 236 páginas.

3.5.2 *Grouper*

Grouper es la interfaz del meta-buscador HuskySearch desarrollado por la Universidad de Washington [Kummamuru et al. 2004]. *Grouper* recupera los resultados de diferentes buscadores en línea y los coloca en grupos etiquetados por frases que describen el contenido del grupo. Se basa en MetaCrawler[2] [URL02]. Su pantalla principal se muestra en la Figura 3.4.

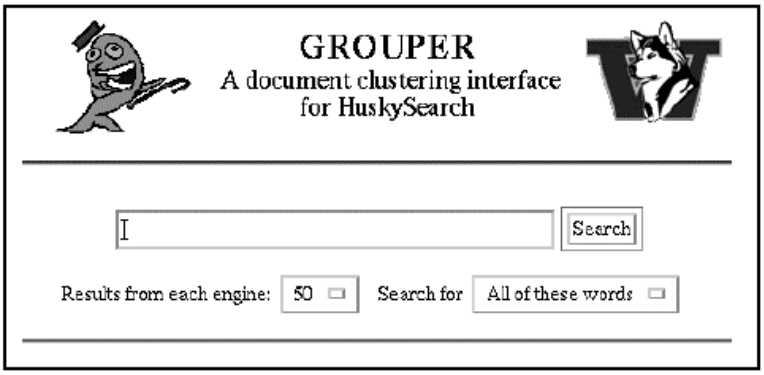


Figura 3.4. Pantalla principal de *Grouper.*

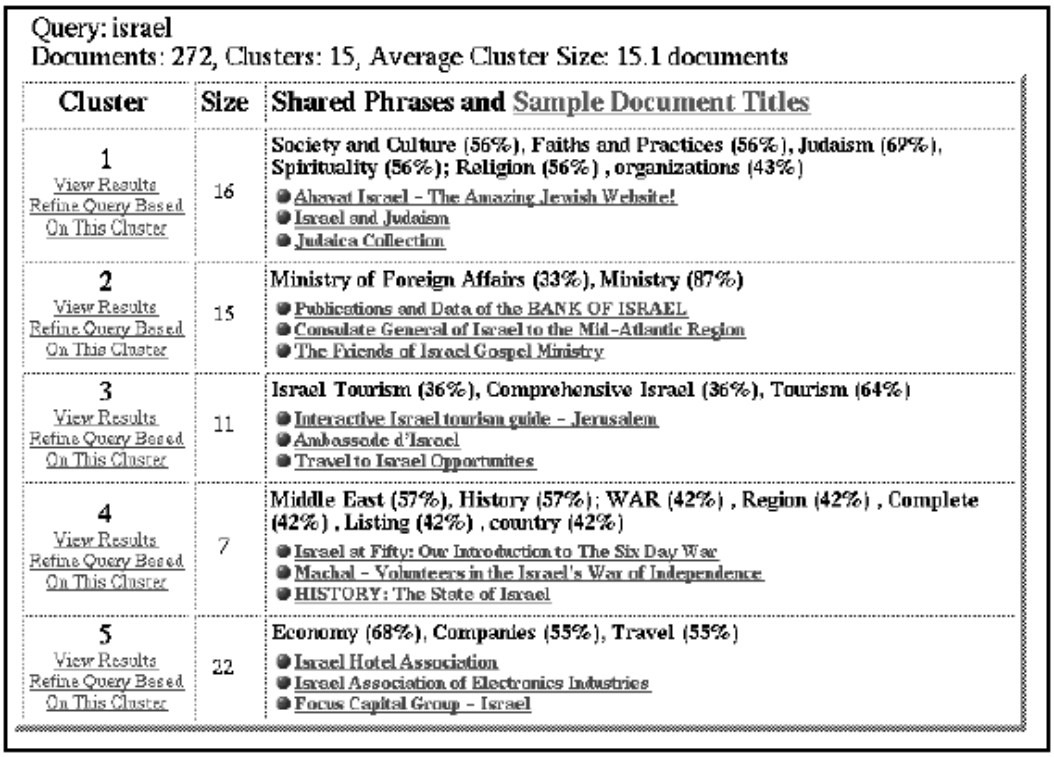
[2] MetaCrawler Web Search Home Page – MetaCrawler, disponible en: http://www.metacrawler.com

Para crear los grupos aplica el algoritmo STC (Suffix Tree Clustering), el cual encuentra frases comunes en los grupos de documentos [Oren & Oren 1999]. STC define como grupo base a un conjunto de documentos que comparten frases comunes. Consta de tres pasos principales:

1. *Limpieza de documentos*.- Se eliminan las palabras vacías y se identifican las palabras clave de los documentos. Posteriormente se aplica un algoritmo de lematización
2. *Identificación del grupo base usando un árbol de sufijos*.- Se crea un índice de frases invertidas para un documento. A cada grupo base se asigna un valor de acuerdo al número de documentos y al número de palabras contenidas en sus frases. En esta etapa se descartan palabras que aparecen en muchos documentos y palabras vacías para que no contribuyan en la evaluación del grupo base
3. *Unión de los grupos base*.- Se unen los grupos base que tienen un gran número de documentos compartidos en un mismo conjunto de documentos. Esto crea grupos con mayor relación semántica, de manera que documentos en un mismo grupo no compartan frases comunes pero si contenido

Este algoritmo permite que entre grupos existan documentos compartidos. Las frases comunes son una forma de resumir el contenido de los documentos. El algoritmo es eficiente para clasificar documentos web, no requiere un parámetro para indicar el número de grupos.

La interfaz de *Grouper* permite que el usuario especifique el número de términos que se considerarán en la consulta y la cantidad de documentos que se desea recuperar. La página de resultados muestra una lista con los documentos recuperados ordenados de acuerdo a una medida de similitud, el número de documentos recuperados y el número de grupos encontrados. Un ejemplo se muestra en la Figura 3.5.



Query: israel
Documents: 272, Clusters: 15, Average Cluster Size: 15.1 documents

Cluster	Size	Shared Phrases and Sample Document Titles
1 View Results Refine Query Based On This Cluster	16	**Society and Culture (56%), Faiths and Practices (56%), Judaism (69%), Spirituality (56%); Religion (56%) , organizations (43%)** ● Ahavat Israel - The Amazing Jewish Website! ● Israel and Judaism ● Judaica Collection
2 View Results Refine Query Based On This Cluster	15	**Ministry of Foreign Affairs (33%), Ministry (87%)** ● Publications and Data of the BANK OF ISRAEL ● Consulate General of Israel to the Mid-Atlantic Region ● The Friends of Israel Gospel Ministry
3 View Results Refine Query Based On This Cluster	11	**Israel Tourism (36%), Comprehensive Israel (36%), Tourism (64%)** ● Interactive Israel tourism guide - Jerusalem ● Ambassade d'Israel ● Travel to Israel Opportunities
4 View Results Refine Query Based On This Cluster	7	**Middle East (57%), History (57%); WAR (42%) , Region (42%) , Complete (42%) , Listing (42%) , country (42%)** ● Israel at Fifty: Our Introduction to The Six Day War ● Machal - Volunteers in the Israel's War of Independence ● HISTORY: The State of Israel
5 View Results Refine Query Based On This Cluster	22	**Economy (68%), Companies (55%), Travel (55%)** ● Israel Hotel Association ● Israel Association of Electronics Industries ● Focus Capital Group - Israel

Figura 3.5. Interfaz de resultados de *Grouper* [Oren & Oren 1999].

En la Figura 3.5 se emplea una tabla, donde cada fila representa el resumen de un grupo que incluye el número de documentos que contiene el grupo, las frases compartidas y algunos títulos. El número entre paréntesis representa el porcentaje de documentos que contienen esa frase. El resumen ayuda al usuario a valorar si el grupo es de su interés. El usuario puede refinar la consulta y aplicar el algoritmo sobre un solo grupo.

Una vez analizado sistemas con funcionalidad similar a CREADOC, el capítulo siguiente describe su diseño.

Capítulo 4. Análisis y diseño de CREADOC

En las siguientes secciones se describen los requerimientos de CREADOC y algunos elementos de diseño. Así también se presenta la estructura funcional de las clases del sistema, los casos de uso de acuerdo al estándar UML y el diseño de la base de datos.

4.1 Análisis del sistema

El análisis del sistema comprende la descripción de los conceptos implementados en el sistema, los cuales definen su funcionalidad, la descripción de los requerimientos funcionales del sistema y del usuario.

Hoy en día, los usuarios realizan búsquedas de documentos utilizando las herramientas que provee el sistema operativo. Por ejemplo, un usuario en plataforma Windows localiza documentos utilizando la herramienta de búsqueda de Windows la cual permite buscar documentos de acuerdo a dos criterios: búsqueda por palabra o frase contenidas en un archivo o por el nombre del mismo.

La *búsqueda por nombre de documento*, es viable cuando el usuario recuerda todo o parte de los términos contenidos en el mismo. El caso de la *búsqueda por contenido* es un proceso complicado cuando se buscan términos contenidos en muchos documentos, debido a que el resultado de la búsqueda es una lista de documentos, la cual no refleja qué documentos podrían contener en mayor grado los términos buscados por el usuario.

4.1.1 Descripción general

El sistema CREADOC implementará las siguientes actividades:

1. *Clasificación:* actividad también denominada agrupamiento. Los usuarios del sistema CREADOC podrán ver la clasificación actual o bien generar una nueva clasificación de sus documentos
2. *Recuperación*: buscará y recuperará los documentos relevantes. Utilizará dos enfoques, el primero es la *búsqueda por contenido*, el cual consiste en localizar los documentos relevantes de acuerdo a una función de similitud entre la consulta y los documentos. El segundo es la *búsqueda por metadatos*, que involucra la búsqueda de los términos en los datos descriptivos del documento como palabras clave, título y autor

4.1.2 Definiciones, abreviaturas y acrónimos

En la especificación de requerimientos se utilizan definiciones, abreviaturas y acrónimos cuyo significado se lista a continuación.

- *CREADOC:* Clasificación y Recuperación Automática de Documentos, sistema que implementa técnicas de clasificación y recuperación de documentos sin supervisión humana
- *Frequent Item-based hierarchical Clustering (FIHC)*, es un algoritmo de agrupamiento de documentos cuyo significado se traduce como *Agrupamiento basado en conjunto de objetos frecuentes*
- *Inverse Document Frequency* (IDF): o Frecuencia Inversa de Documentos, formalmente es definido como el logaritmo del conciente entre el número total de documentos y el número de documentos que contienen el término buscado
- *Palabras claves:* conjunto de términos que describen el contenido de un documento. Generalmente son los términos con mayor frecuencia de aparición dentro del documento
- *Palabras vacías:* llamadas en Inglés *stopwords,* son un conjunto de términos no significativos (tales como artículos, preposiciones, adjetivos), que se consideran irrelevantes al realizar búsquedas
- *Retrieval Information (RI):* en Español, recuperación de información
- *Soporte global mínimo (GS)*: porcentaje mínimo de documentos en que deberá aparecer un término para que se considere término frecuente
- *Soporte grupal mínimo (CS)*: el porcentaje mínimo de aparición de un término en un grupo de documentos
- *Term Frequency (TF)*, Frecuencia de un término, número de apariciones de un término en un documento
- *Usuario*: cualquier persona que interactúe con el sistema

4.1.3 Requerimientos funcionales

1. *Clasificación*
   1.1. *Clasificación actual*. Mostrar la estructura de grupos formada durante la última clasificación. Desplegar los parámetros que generaron dicha clasificación seguido de la etiqueta de cada grupo y sus documentos respectivos
   1.2. *Clasificación nueva*. Especificar los porcentajes de similitud entre los documentos y grupos de documentos, (soporte grupal y soporte global respectivamente). Una vez dados estos parámetros, aplicar el algoritmo de clasificación FIHC. Al finalizar la clasificación de los documentos, mostrar la estructura de clases generada

2. *Recuperación* (o búsqueda de documentos relevantes)
   2.1. *Búsqueda por metadatos o búsqueda básica*. Para realizar esta búsqueda es necesario que el usuario elija un parámetro de búsqueda como palabras clave, título o autor e incluya los términos a buscar dentro del parámetro seleccionado. Seleccionar si se requiere encontrar todos los términos introducidos o sólo algunos
   2.2. *Búsqueda por contenido*: Introducir los términos a buscar dentro de los documentos, la similitud mínima entre el documento y la consulta y seleccionar los grupos sobre los cuales desea realizar la consulta. Por defecto, la búsqueda se realizará en todos los documentos
3. *Agregar documento*: Para que un documento pueda clasificarse se requiere que esté dado de alta en la base de datos. El usuario debe introducir datos descriptivos del documento como son título, autor, idioma, palabras clave y ubicación del archivo

4.1.4 Requerimientos no funcionales

- El sistema no almacenará los documentos, por lo que no permitirá visualizar el contenido de los mismos
- El sistema no analizará documentos que no estén en formato texto
- El sistema no funcionará como acceso a un servidor remoto de documentos

4.1.5 Requerimientos del usuario

Las herramientas de administración deben cumplir con ciertas características como son:

1. *Facilidad de uso.*- El sistema contará con una interfaz que permita al usuario manejar el sistema con facilidad
2. *Rapidez*.- El sistema realizará las tareas que le indique el usuario en el menor tiempo posible
3. *Interfaz amigable*.- El sistema tendrá una interfaz que facilite la comprensión de las actividades a realizar
4. *Interacción sencilla*.- El sistema no requerirá que los usuarios aprendan un nuevo lenguaje o sintaxis para expresar las consultas; se utilizará lenguaje natural

4.1.6 Requerimientos del sistema

El programa funcionará en cualquiera de los sistemas operativos con soporte para Java, por ejemplo: Windows 95, 98, ME, NT4, 2000 y XP, Linux y Solaris.

El sistema se desarrollará utilizando las siguientes aplicaciones:

- Un servidor de base de datos MySQL versión 5.0.1
- Un contenedor de Servlets, Apache Tomcat versión 5.5
- El JRE (Java Runtime Enviroment) o máquina virtual de Java versión 1.4

Cabe señalar que estos programas pertenecen a la categoría de software libre, por lo que no se requiere la compra de licencias para su utilización.

4.2 Diseño del sistema

La arquitectura de CREADOC está dividida en cuatro subsistemas: Clasificar, Recuperar, Agregar Documento y Ayuda, como se ilustra en la Figura 4.1.

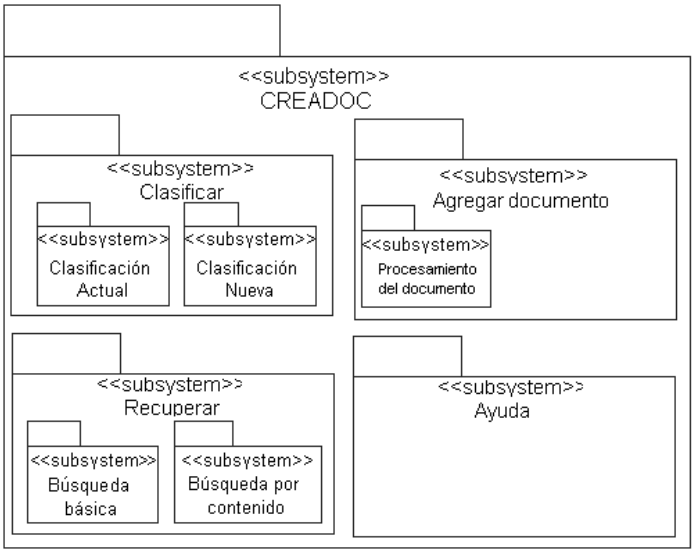


Figura 4.1. Arquitectura del sistema CREADOC**.**

A continuación se describe cada subsistema.

4.2.1 Subsistema de clasificación

El subsistema clasificar estará integrado de dos subsistemas, uno para generar nuevas clasificaciones y otro para mostrar la clasificación actual.

a) *Clasificación nueva*.- Este subsistema contendrá todos los módulos que permiten crear grupos de documentos. Para generar una nueva clasificación el usuario proporcionará los valores de soporte global y soporte de grupo. El sistema mostrará la estructura de las clases generadas

b) *Clasificación actual*.- Este subsistema se encargará de acceder a la clasificación existente y mostrará la información de los grupos

4.2.2 Subsistema de recuperación

La recuperación de documentos podrá hacerse a nivel de contenido y metadatos.

a) *Búsqueda por contenido*.- Este módulo permitirá acceder a la información relevante de los documentos y buscará sólo los que tengan mayor similitud con la consulta del usuario. Para esta consulta el usuario especificará los términos que desea buscar, la similitud mínima entre los documentos y la consulta y seleccionará las etiquetas de los grupos sobre los que se desea que se realice la búsqueda.

b) *Búsqueda por metadatos*.- Este tipo de búsqueda se realizará en todos los documentos y comparará los términos de la consulta con los que forman los descriptores del documento. Para realizar una consulta de este tipo, el usuario seleccionará el descriptor sobre el que quiere buscar, introducirá su consulta y señalará si desea que los documentos recuperados contengan todos o sólo algunos de los términos.

4.2.3 Subsistema de agregación de documento

Este subsistema permitirá agregar un documento a la colección. Para agregar un documento será necesario introducir la información descriptiva del documento como título, autor, idioma y palabras clave. Además, en este módulo se extraerán los términos relevantes del contenido del documento.

4.3 Casos de uso

Para el sistema CREADOC se identificaron los casos de uso de la Figura 4.2. Este diagrama muestra la interacción entre cada una de las actividades que va a desempeñar el usuario en el sistema.

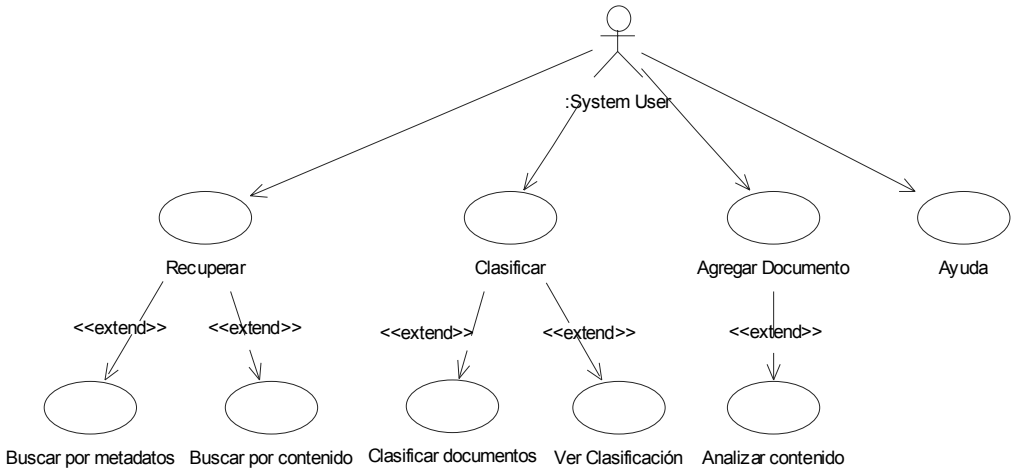


Figura 4.2. Diagrama de casos de uso.

4.3.1 Caso de Uso 1: Buscar por contenido

*Actores***:** Usuario, (cualquier persona que ingrese al sistema).
Descripción: El usuario entra al sistema y selecciona la opción *Búsqueda por contenido*.

Acción del actor	**Sistema**
1.- Este caso de uso comienza cuando el usuario ha abierto la página de inicio del sistema y selecciona la opción realizar *búsqueda por contenido*.	
	2.- El sistema muestra al usuario otra página donde solicita los términos a buscar, el porcentaje de similitud y las etiquetas de los grupos para que el usuario tenga la opción de limitar su espacio de búsqueda.
3. El usuario introduce los términos a buscar, selecciona el porcentaje de similitud mínima entre los documentos y los grupos y marca los grupos sobre los que desea realizar la búsqueda.	
	4.-. El sistema realiza la búsqueda de documentos que cumplan con los parámetros especificados por el usuario.
5.-El usuario visualiza la lista de documentos que cumplieron con su consulta.	

Flujos alternativos:

- En el punto 3, si el usuario no selecciona un porcentaje de similitud mínimo, se aplicará el valor que se imprime por defecto. En el caso de los grupos, si el usuario no cambia o no selecciona un grupo en particular, la búsqueda se aplicará sobre toda la colección de documentos.

En la Figura 4.3 se muestra el diagrama de secuencia para el caso de uso: *buscar por contenido*.

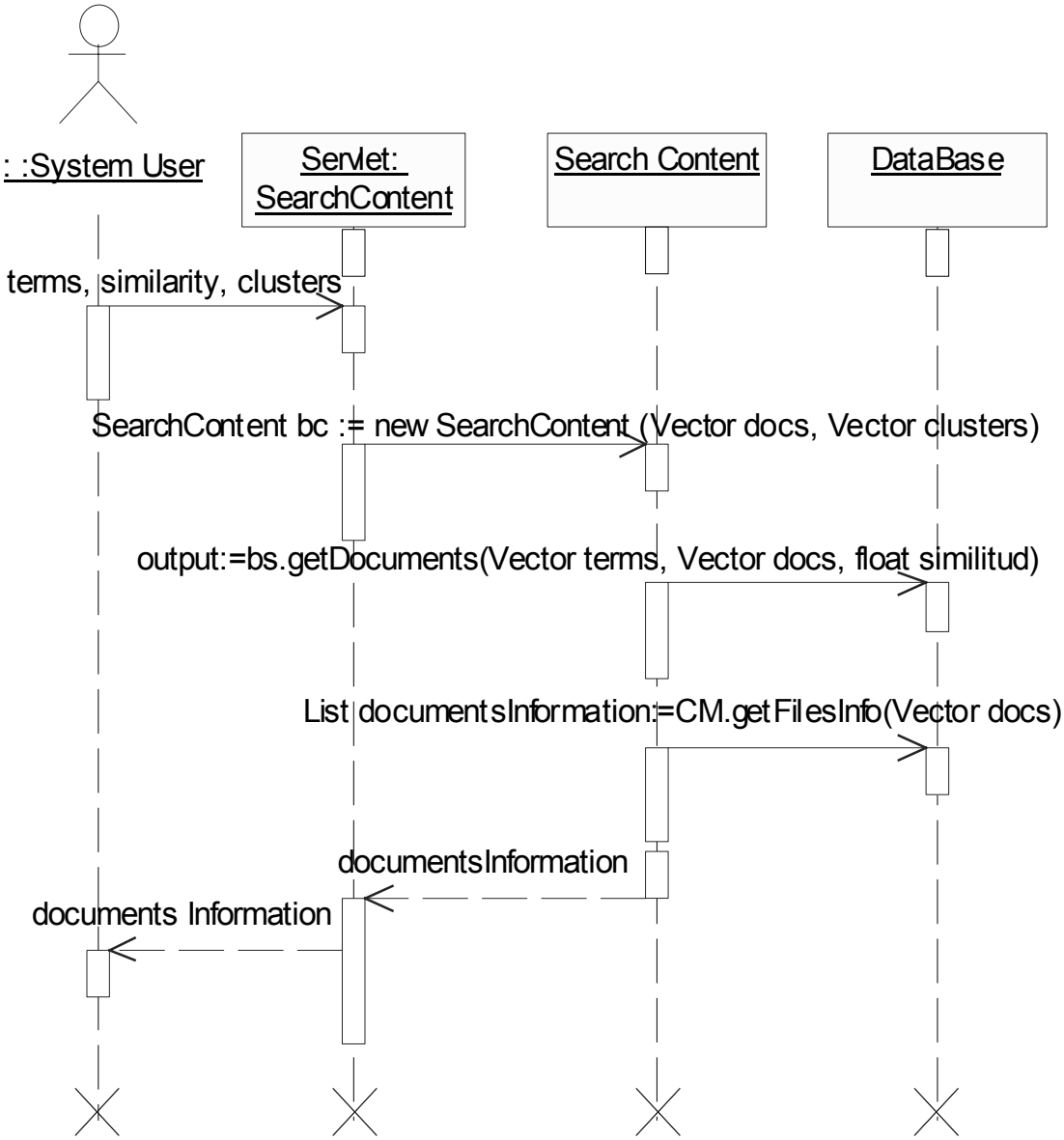


Figura 4.3. Diagrama de secuencia para el Caso de Uso 1: *Buscar por contenido*.

4.3.2 Caso de Uso 2: Buscar por metadatos

Actores: Usuario, (cualquier persona que ingrese al sistema)
*Descripción***:** El usuario entra al sistema y selecciona la opción *Búsqueda básica*.

Acción del actor	**Sistema**
1.- Este caso de uso comienza cuando el usuario ya se encuentra dentro del sistema y selecciona la opción de realizar una *búsqueda básica.*	
	2.- El sistema muestra al usuario una pantalla donde solicita el tipo de búsqueda a realizar, los términos a buscar y la opción de buscar todos los términos o sólo algunos.
3. El usuario selecciona el tipo de búsqueda a realizar, ya sea por título, autor o palabra clave. Introduce los términos a buscar y selecciona si desea que se busquen los documentos que contienen todos o algunos de los términos.	
	5. El sistema realiza la búsqueda de documentos que cumplan con los parámetros especificados por el usuario.
6. El usuario visualiza la lista de documentos que cumplieron con su consulta.	

Flujos alternativos:

- En el punto 3, por omisión, la búsqueda considera todos los términos en las palabras clave.

La secuencia que se lleva a cabo en el caso de uso *buscar por metadatos* se muestra en la Figura 4.4.

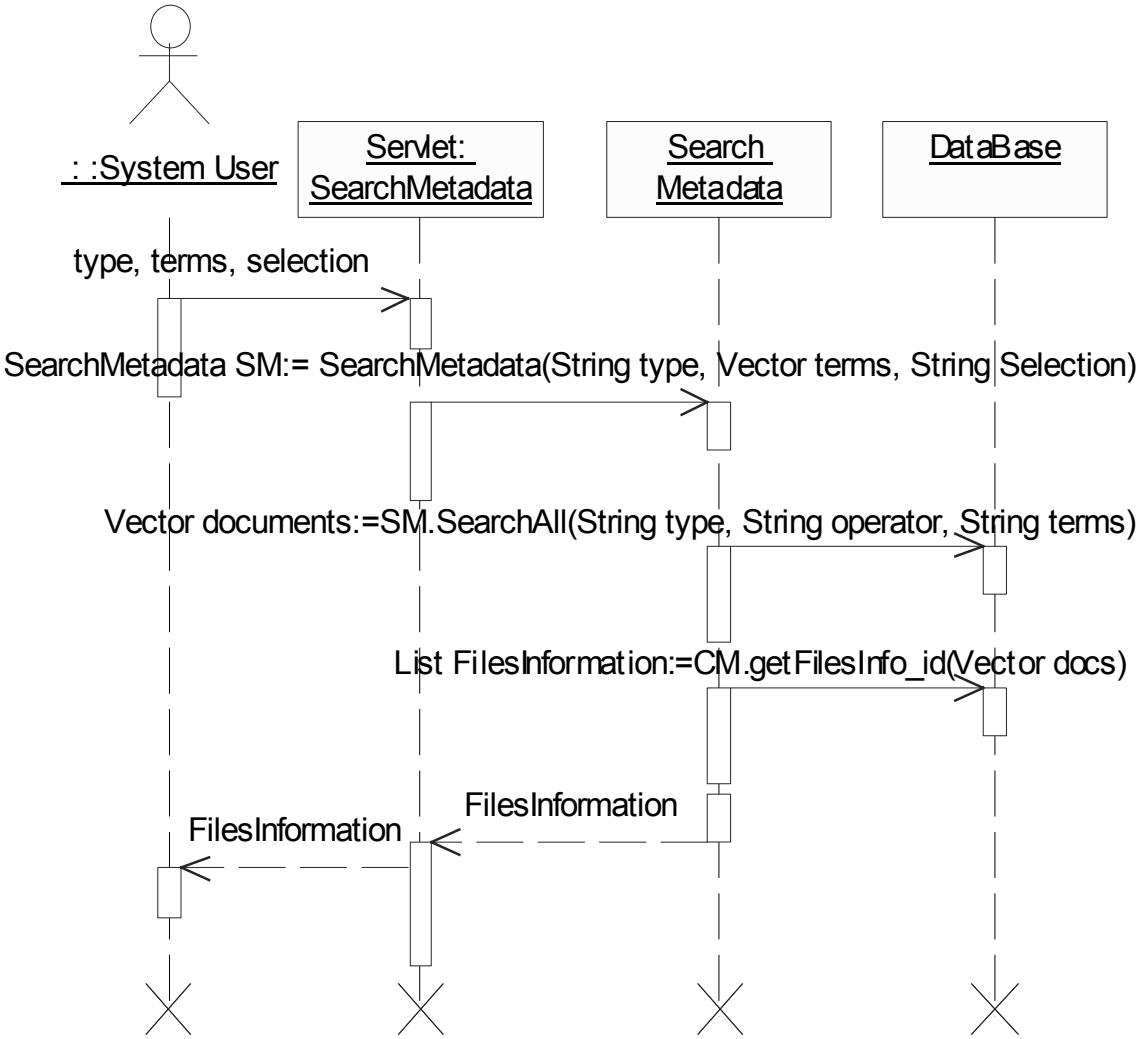


Figura 4.4. Diagrama de secuencia para el Caso de Uso 2: *Buscar por metadatos*.

4.3.3 Caso de Uso 3: Clasificar documentos

Actores: Usuario, (cualquier persona que ingrese al sistema)
Descripción: El usuario accede al sitio y selecciona la opción *Clasificar*. Donde se solicitan los parámetros a considerar para generar la nueva clasificación.

Acción del actor	**Sistema**
1.- Este caso inicia cuando el usuario se encuentra en la página de CREADOC y selecciona la opción de Clasificación nueva.	
	2.- El sistema le solicita el soporte global mínimo y soporte grupal mínimo, datos indispensables para generar nuevos grupos.
3.- El usuario selecciona un valor para el soporte mínimo global y el soporte de grupo.	
	4.- El sistema aplica el algoritmo de agrupamiento al conjunto de documentos, de acuerdo a los valores seleccionados por el usuario.
5.- El usuario visualiza la estructura de grupos, formada por el nombre del grupo, seguido del título, autor y el nombre del archivo de cada documento del grupo.	

Flujos alternativos:

- En el punto 3, si el usuario no selecciona ningún valor se utilizarán los valores que tiene por defecto el sistema.

En la Figura 4.5 se muestra el diagrama de secuencia para el caso de uso *clasificar documentos*.

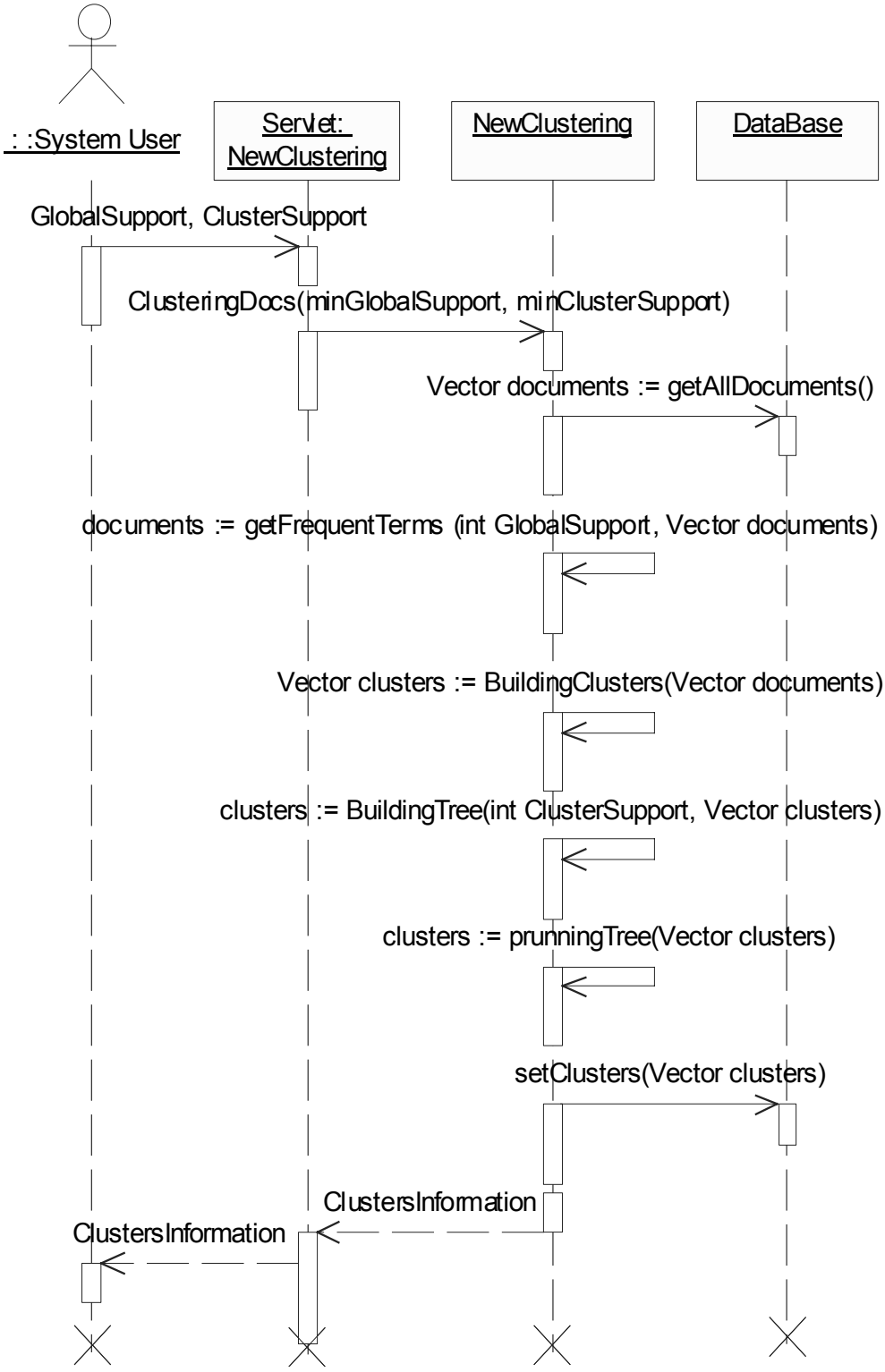


Figura 4.5. Diagrama de secuencia para el Caso de Uso 3: *Clasificar documentos*.

4.3.4 Caso de Uso 4: Agregar documento

Actores: Usuario, (cualquier persona que ingrese al sistema)
Descripción: El usuario entra al sistema y selecciona la opción *Agregar documento*.

Acción del actor	Sistema
1.- Este caso de uso comienza cuando el usuario ya se encuentra dentro del sistema y selecciona la opción agregar documento.	
	2.- El sistema muestra un formulario solicitando los descriptores del documento y nombre del archivo que contiene el documento.
3. El usuario introduce el título del documento, los autores, el idioma en que está escrito, las palabras que describen al documento y la ubicación del archivo.	
	5. El sistema valida los datos, analiza el documento y lo almacena en la base de datos.
6. El usuario visualiza una página donde se le confirma que sus documentos han sido agregados a la colección de documentos.	

Flujos alternativos:

- En el punto 3, si no fueron introducidos los parámetros obligatorios se mostrará un mensaje de error, de igual manera si el archivo seleccionado no es de texto.

En la Figura 4.6 se muestra el diagrama de secuencia para el caso de uso: *Agregar documento*.

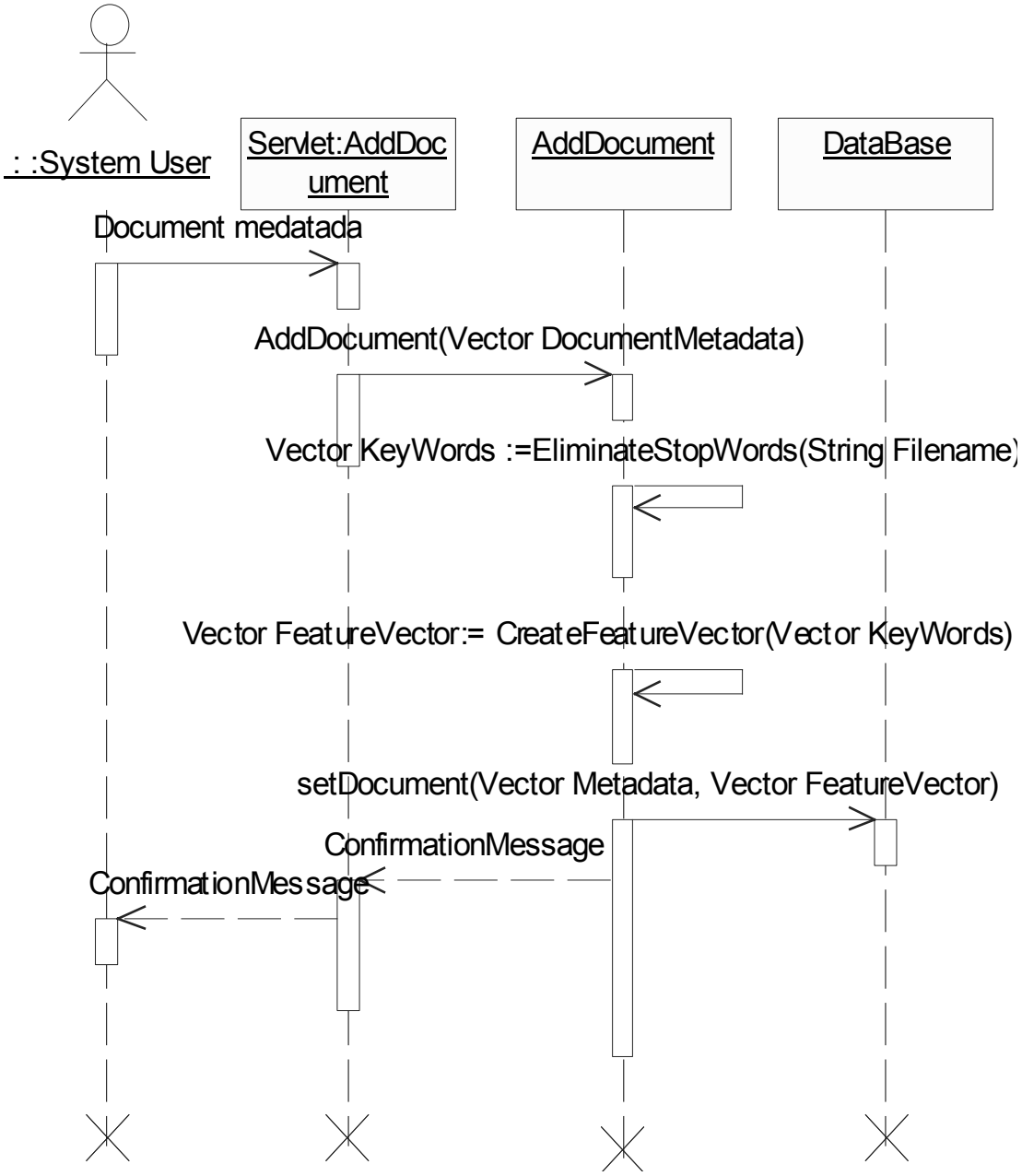


Figura 4.6. Diagrama de secuencia para el Caso de Uso 4: *Agregar documento.*

4.4 Análisis y diseño de la base de datos

Para llevar un control de los documentos, analizarlos y manejar la información contenida en ellos se hace uso de una base de datos, la cual se describe en los puntos siguientes.

4.4.1 Modelo conceptual

El sistema CREADOC trabaja con el contenido de documentos, cuya base de datos cumple con las siguientes características:

Para cada documento se conoce su identificador único (ID_DOCUMENT), el título (TITLE), el idioma en que está escrito (LANGUAGE) y las palabras clave (KEYWORDS) que describen el contenido del documento. Además, se incluye información de su autor o autores.

En el caso de los autores se requiere conocer su apellido o apellidos (LASTNAME1 y LASTNAME2), su nombre(s) (NAME) y un identificador único de autor (ID_AUTHOR). El segundo apellido es opcional.

El sistema no almacena los documentos en su totalidad, sólo los términos descriptivos y palabras relevantes (WORDS). Para cada palabra relevante (WORD) se requiere conocer la frecuencia máxima de aparición en el conjunto de documentos (FMAX), la frecuencia inversa del término en un documento (IDF) cuyo valor se calcula a partir de la frecuencia máxima y la frecuencia de la palabra en el documento (TF).

4.4.2 Diseño de la base de datos

La Figura 4.7 muestra el diagrama entidad-relación del sistema CREADOC.

La entidad WORDS es una entidad débil ya que forzosamente se requiere que exista un documento para que exista una palabra.

La base de datos cuenta con un atributo derivado, IDF, el cual se obtiene a partir de los valores contenidos en FMAX y TF.

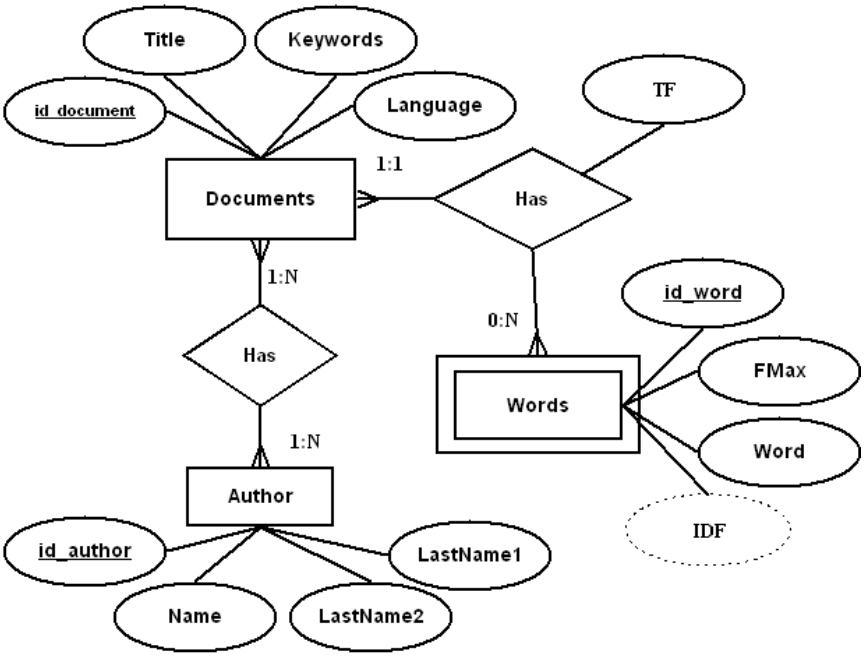


Figura 4.7. Diagrama entidad relación de la base de datos

La Figura 4.8 muestra la estructura de la base de datos en términos de relaciones y atributos. Cabe mencionar que de acuerdo a las especificaciones del sistema no es necesaria la relación *Documents_words_temp*, sin embargo el sistema hace uso de una herramienta denominada Hermes que realiza la recuperación de información sobre esta relación, la cual contiene únicamente los identificadores de los documentos de los grupos seleccionados por el usuario, los términos descriptivos de los documentos y junto con su frecuencia. El capítulo 5 contiene la descripción de Hermes.

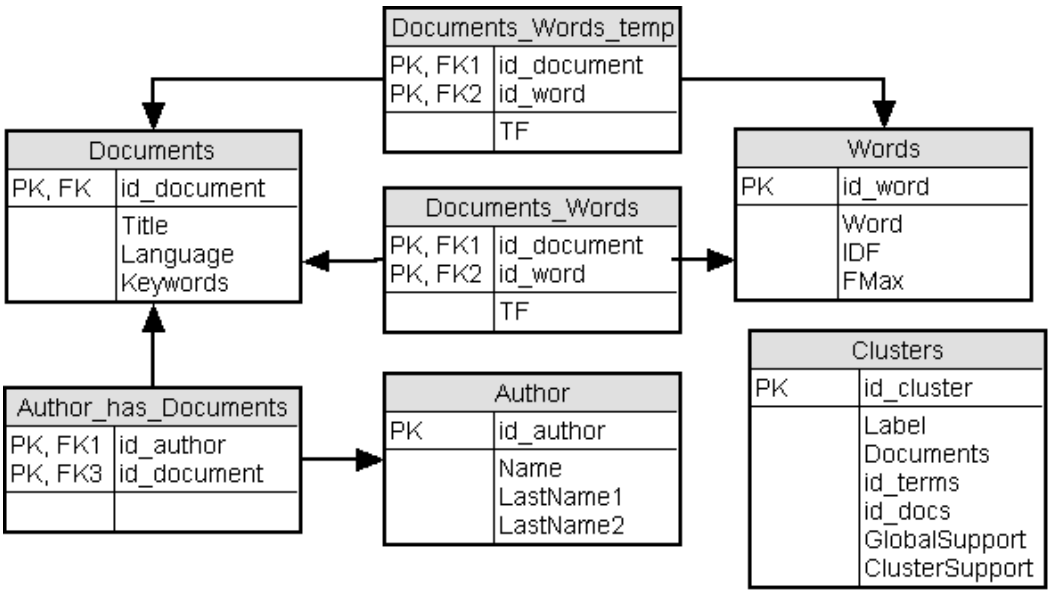


Figura 4.8. Diseño de la base de datos

4.4.3 Diccionario de datos

En las relaciones siguientes se describen los parámetros más importantes de la base de datos y los tipos de datos soportados.

La relación *Documents* contiene la información principal acerca de los documento, analizados en el sistema.

Relación Documents				
Atributos	**Tipo**	**Null**	**key**	**Descripción**
id_document	int (11)	No	PK	Identifica a un documento
Title	Varchar(150)	No		Título del documento
Language	enum('spa', 'eng')	Si		Lenguaje del documento
Keywords	Enum	No		Palabras que describen el contenido del documento.

La relación *Words* contiene la información referente a los términos o palabras contenidas en los documentos.

Relación Words				
Atributos	**Tipo**	**Null**	**key**	**Descripción**
id_word	int (11)	No	PK	Identifica a un término
Word	Varchar(50)	No		Palabra
IDF	float (9,3)	No		Frecuencia Inversa de Documentos
FMAX	int (11)	No		La frecuencia máxima de aparición del término en el conjunto de documentos

En la relación *Autor*, se almacenan los datos personales de los autores de los documentos.

Relación Autor				
Atributos	**Tipo**	**Null**	**key**	**Descripción**
id_author	int (11)	No	PK	Identifica a un autor
Name	Varchar(50)	Si		Nombre del autor
LastName1	Varchar(50)	Si		Primer apellido del autor
LastName2	Varchar(50)	Si		Segundo apellido del autor.

La relación *Clusters* se utiliza para almacenar el resultado del agrupamiento, en ella se resume el contenido de cada grupo.

Relación Clusters				
Atributos	**Tipo**	**Null**	**key**	**Descripción**
id_cluster	int (11)	No	PK	Identifica a un grupo
label	text	No		Nombre del grupo
documents	Text	No		Títulos de los documentos contenidos en el grupo
id_terms	text	No		Identificadores de términos que describen al grupo
id_docs	text	No		Identificadores de documentos del grupo
GlobalSupport	int (3)	Sí		Soporte global utilizado para generar al grupo
ClusterSupport	int (3)	Sí		Soporte de grupo utilizado para generar la clasificación

La relación entre documentos y los términos contenidos en cada documento se modela en la tabla *Documents_Words*.

Relación Documents_Words				
Atributos	**Tipo**	**Null**	**Key**	**Descripción**
id_document	int (11)	No	MUL	Identifica a un término
id_word	int (11)	No	MUL	Identifica a una palabra
TF	int (11)	Sí		Frecuencia de aparición del término con identificador id_word en el documento identificado por id_document

Documents_Words_temp relaciona los documentos y términos contenidos en los documentos, se utiliza como una relación temporal para realizar búsquedas en grupos de documentos.

Relación Documents_Words_temp				
Atributos	**Tipo**	**Null**	**Key**	**Descripción**
id_document	int (11)	No	MUL	Identifica a un término
id_word	int (11)	No	MUL	Identifica a una palabra
TF	int (11)	Sí		Frecuencia de aparición del término id_word en el documento id_document

La relación entre los autores y los documentos se representa en la relación *Author_has_Documents*.

Relación Author_has_Documents				
Atributos	**Tipo**	**Null**	**key**	**Descripción**
id_document	int (11)	No	MUL	Identifica a un documento
id_author	int (11)	No	MUL	Identifica a un autor

Una vez creado el diseño de cada uno de los módulos a utilizar en CREADOC, se requiere conocer los detalles de implementación, los cuales se muestran en el siguiente capítulo.

Capítulo 5. Implementación

CREADOC fue diseñado con un paradigma orientado a objetos lo que permite que algunos módulos funcionen de manera independiente. El sistema hace uso de Hermes, una herramienta para recuperar documentos, la cual se explica en la sección 5.3. Se implementó el algoritmo FIHC como técnica de clasificación de documentos y se hizo uso de métodos para procesar consultas en lenguaje natural. Los detalles de implementación se describen en las secciones siguientes.

5.1 Descripción del sistema

Para apoyar la administración de una colección de documentos, CREADOC implementa las tareas principales siguientes: agrupamiento y recuperación de información. A diferencia de la recuperación de información, para el agrupamiento el usuario puede modificar los parámetros de similitud entre los elementos de un grupo o entre grupos de documentos. En tanto, la recuperación puede realizarse con base en el contenido o mediante elementos descriptivos del documento. El sistema trabaja con documentos de texto debido a que la conversión de otros formatos a éste es un procedimiento incorporado por la mayoría de los editores de texto.

Con el objetivo de facilitar el manejo del sistema, en la interfaz se emplea el término *clasificación* en lugar de *agrupamiento* considerando que el agrupamiento es un tipo de clasificación. A su vez, se utiliza el término *búsqueda* para hacer referencia a una tarea de *recuperación de información*.

5.2 Características de la implementación

CREADOC se implementó bajo la plataforma Java, la cual consta de un lenguaje de programación orientado a objetos, una máquina virtual o JRE que permite la portabilidad y una biblioteca estándar o API[3]. El lenguaje permite el desarrollo de programas con interfaz gráfica o textual, que pueden incrustarse o cooperar con los navegadores web. En CREADOC, la interfaz web emplea Servlets, los cuales se describen en la sección 5.2.1.

El sistema se desarrolló con un procesador Intel Pentium IV a 2.5 GHZ, con 256 MB de RAM, sin embargo en las fase de pruebas se ejecutó en un procesador Athlon XP 2400 con 512 de RAM,

[3] API, *Application Programming Interface* - Interfaz de Programación de Aplicaciones

mostrando un desempeño favorable. El espacio en disco duro necesario dependerá del número de documentos que el usuario registre en el sistema.

5.2.1 Uso de Servlets

Los acrónimos Servlet y applet emplean el sufijo *let* a manera de diminutivo para indicar que se trata de programas pequeños. Un *applet* se refiere a un programa pequeño escrito en Java que se ejecuta en un navegador web. Por contraposición, un Servlet se ejecuta en un servidor web. Los Servlets son objetos que extienden su funcionalidad al utilizar el contexto de un contenedor de Servlets, (conocido también como servidor de aplicaciones). Tomcat es un ejemplo de un contenedor de este tipo.

CREADOC utiliza Servlets para generar páginas web de forma dinámica. Esta funcionalidad requiere de parámetros que se envían a manera de solicitudes utilizando el protocolo HTTP a través del navegador web. Si los parámetros y su valor se almacenan en un archivo con extensión *properties*, cuando se desee modificar el valor de uno o todos los parámetros, no será necesario un proceso de compilación. Generalmente los archivos con esta extensión se conocen como *archivos de configuración*.

Los applets y Servlets constituyen tecnologías ampliamente utilizadas en el desarrollo de aplicaciones en web. En CREADOC se eligieron los Servlets debido a que el tiempo de ejecución es menor al de un applet. Por otro lado, la combinación de Servlets con archivos de configuración puede incrementar la flexibilidad al facilitar modificaciones en la interfaz. El archivo de configuración de CREADOC se denomina *localStringsC.properties*. A continuación se enlistan los elementos que pueden modificarse.

- Mensajes, títulos e instrucciones en todas las páginas a excepción de la descripción del sistema. Se asume que la descripción es estática puesto que contiene los objetivos principales de CREADOC
- Tipo, tamaño y color de fuentes de todos los mensajes e instrucciones
- Color de fondo y color de líneas
- Mensajes y colores del menú
- Logotipo e imágenes
- Correo electrónico y vínculo de contacto

Para realizar los cambios, basta con que el usuario modifique el archivo de configuración y reinicie el contenedor de Servlets. Este archivo está estructurado de acuerdo a la funcionalidad de las páginas. Cada una de sus líneas representa un par parámetro – valor. El apéndice B contiene una descripción detallada del contenido del archivo *localStringsC.properties*.

La recuperación de información utiliza Hermes, el cual hace uso de los metadatos establecidos por la DCMI. Debido a que CREADOC está orientado a usuarios que conocen su colección de documentos, no se consideraron todos los elementos, sólo los siguientes: el título del documento, el nombre de sus creadores o autores, el idioma y el identificador único del documento.

5.2.2 Procesamiento de textos

Para facilitar al usuario la búsqueda en el sistema y la introducción de palabras clave el usuario escribe las consultas en lenguaje natural. El sistema se encarga de hacer un procesamiento previo de la consulta y los documentos eliminando palabras vacías tales como artículos, pronombres, preposiciones, signos de puntuación y unificando las palabras a letras minúsculas.

5.3 Módulo de recuperación de información

CREADOC implementa el módulo de recuperación de información con Hermes, el cual es un componente que implementa diferentes modelos de recuperación de información, a saber, modelo booleano extendido, espacios vectoriales y semántico latente [Maldonado 2002]. El sistema en particular, utiliza el modelo de espacios vectoriales. Ver el capítulo 2 para consultar las características de este modelo.

La comunicación con Hermes se puede realizar de dos maneras, la primera emplea *RMI* (*Remote Method Invocation*), la cual es una implementación en java que permite una comunicación distribuida entre objetos, facilita la localización de un objeto dentro del servidor, el llamado de sus métodos, el envió de parámetros y la recepción de la respuesta. La segunda utiliza la librería *IRServer.jar*, que permite efectuar el proceso de RI de manera local. CREADOC emplea la segunda alternativa para mantenerse independiente del estado del servidor de Hermes.

Para que el sistema reconozca a la librería *IRServer.jar*, es necesario especificar la ruta en las variables del entorno. El proceso de recuperación inicia con el método OpenConnection() que regresa el número de conexión y reserva recursos; concluye con una llamada al método CloseConnection (int i).

La arquitectura de Hermes permite utilizar alguno de estos modelos sin modificar la aplicación. Para ello requiere que la aplicación y colecciones cumplan cierta estructura.

La interacción con Hermes emplea los parámetros siguientes: una consulta, un grado de similitud, la ruta de acceso a la colección y el modelo de recuperación que se desea implementar. Como resultado se obtienen los metadatos de los documentos relevantes y el grado de similitud entre éstos y la consulta. Dichos metadatos se basan en el estándar Dublin Core.

Hermes cuenta con métodos para acceder a las colecciones y obtener sus datos. Para acceder a una colección se debe extender la clase abstracta *DocumentCollection* que permite la extensión y definición de métodos. Es importante que se especifiquen los valores de recuperación de información (*setValues*), los términos y las fuentes de información. Después de especificar estos valores, el método *getRelevantDocuments* hace uso de los siguientes métodos para acceder a las colecciones:

1. *getFromCollectionTfIdf.* Busca los documentos que tengan al menos un término de la consulta y recupera la frecuencia y frecuencia inversa de los términos de la consulta.
2. *getFromCollectionTfAndTerms*. Busca los documentos que tengan al menos un término de la consulta y recupera la frecuencia, frecuencia inversa y términos que describen a estos documentos.

Los métodos retornan la información de los documentos relevantes en un objeto tipo DocumentData. El método *getMetaData* regresa los metadatos de un objeto *DocumentData*. CREADOC implementa el acceso a la colección mediante el uso de las relaciones *Documents_Wors_temp y Words* y la obtención de los metadatos de los documentos utilizando *Documents, Author y Autor_has_Documents*.

5.4 Módulo de clasificación de documentos

Para disminuir el tiempo de la recuperación de información, se implementó un módulo que permite clasificar los documentos. Este módulo implementa el algoritmo FIHC descrito en el capítulo 3.

El funcionamiento del algoritmo FIHC requiere que se proporcione el conjunto de términos frecuentes. CREADOC realiza una eliminación de las palabras vacías y un conteo de los términos que aparecen en el documento. La selección de los términos que describirán al documento en el vector característico se hace de dos maneras. La primera consiste en especificar en el archivo de configuración el número mínimo de veces que se requiera aparezca un término en el documento. Por ejemplo, se desea que el vector característico esté formado por términos que se repitan en el documento mínimo 30 veces, entonces se asignan los valores siguientes: *DI.minOmax=**min** y DI.value=**30***. Cuando a *DI.minOmax* se le asigna el valor max, entonces representa el tamaño del vector característico, por ejemplo, se puede especificar que el vector característico de cada documento contenga 20 términos al asignar los valores siguientes: *DI.minOmax=max y DI.value=20.* Por omisión, las palabras claves

asignadas por el usuario al agregar un documento se consideran términos frecuentes cuyo valor de frecuencia es igual al máximo valor de frecuencia del resto de términos.

Las fases de FIHC se encuentran codificadas en una estructura de clases como se muestra en la Figura 5.1.

Un grupo se representa mediante la clase *Cluster*. Cada objeto *Cluster* tiene un nombre (*name*), el cual consiste de una cadena compuesta por los identificadores de términos frecuentes (*FrequentItem*) que describen al grupo. Contiene además, los identificadores de los términos (*id_words*) y documentos (*id_documents*), el soporte de cada uno de los términos frecuentes (*ClusterSupport*), los documentos contenidos en los grupos que se derivan del objeto *Cluster* (*DocsOfSubclusters*) y los términos que se combinaron para formar al grupo (*Parents*).

La clase *BuildingClusters*, construye grupos formados por documentos, el nombre de cada grupo está compuesto por los identificadores de los términos que describen a los documentos. El primer paso es obtener todos los términos frecuentes (*GlobalFrequentItem*), seguido de los conjuntos de objetos frecuentes (*GlobalFrequentItemSet*), esta etapa es la implementación del algoritmo *apriori*. Después se asigna cada documento a un único grupo (*DisjointClusters*), y se recalcula el soporte de cada uno de los grupos formados (*CalculateClusterSupport*).

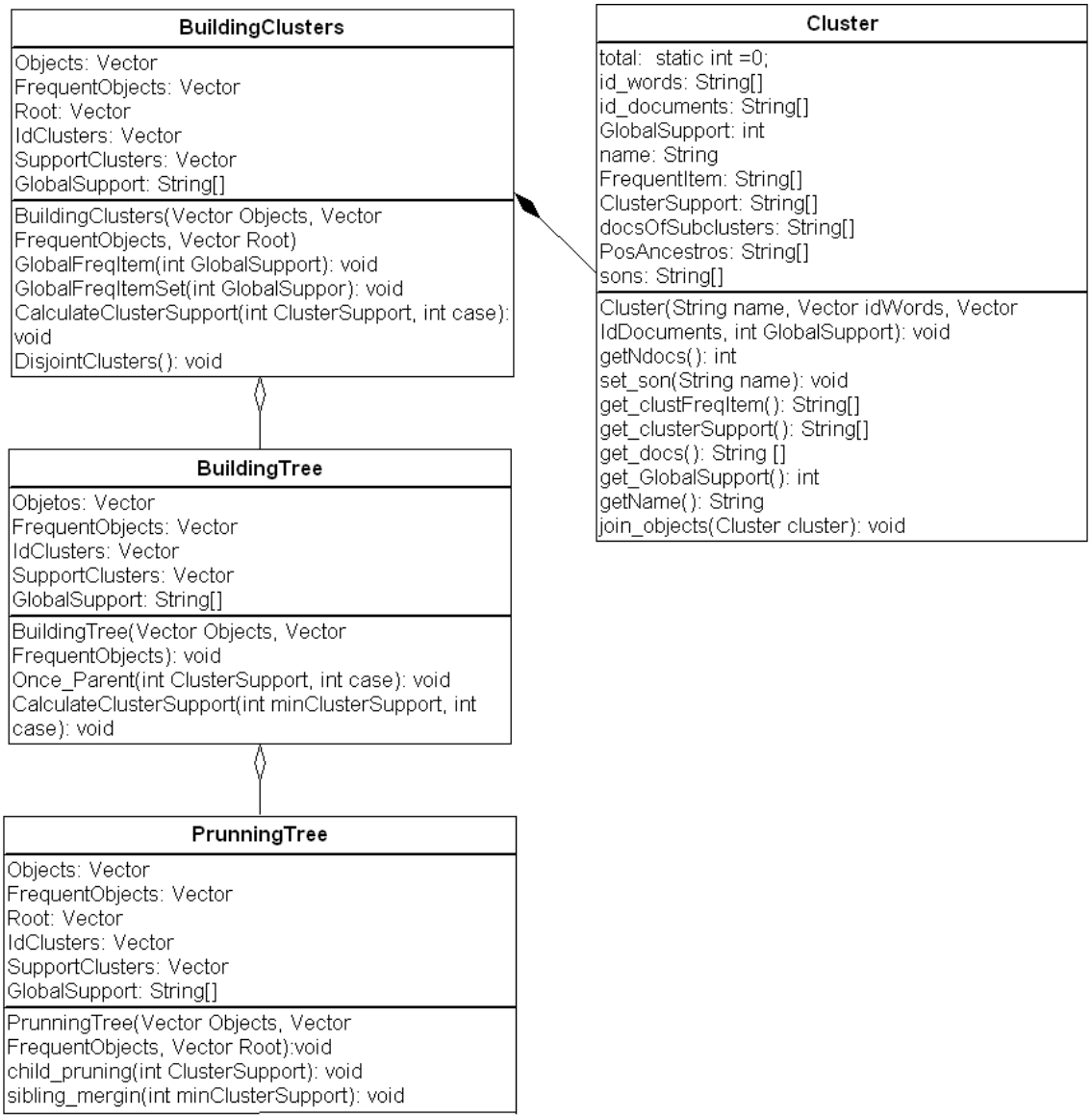


Figura 5.1. Estructura de clases de la implementación del algoritmo FIHC.

La segunda fase de FIHC se implementa en la clase *BuildingTree*, la cual construye una estructura de árbol, asignando un único antecesor a cada grupo (*Once_Parent*). Debido a los cambios generados, se recalcula el soporte de cada grupo (*CalculateClusterSupport*).

Finalmente se requiere podar el árbol de grupos, primero en profundidad (*child_prunning*) y después en anchura (*sibling_merging*). Ambas podas están implementadas, sin embargo la poda a lo ancho no se ejecuta en la versión final, debido a que la búsqueda sobre estructuras jerárquicas se ve beneficiada con árboles anchos.

Para consultar las características del algoritmo de clasificación FIHC, ver el Capítulo 3.

5.5 Descripción de interfaces

CREADOC fue implementado bajo una interfaz web. La Figura 5.2 muestra el mapa del sitio web.

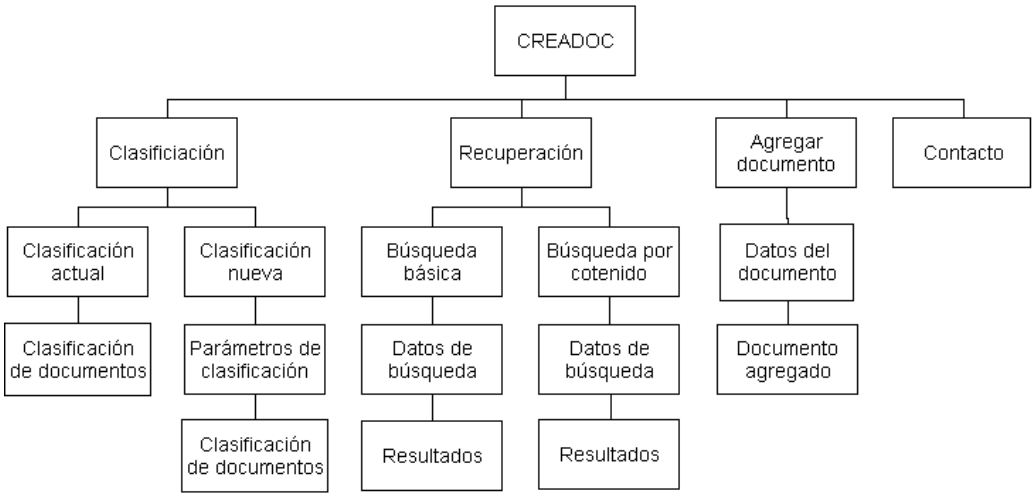


Figura 5.2. Árbol de distribución de las páginas del sitio web.

El sistema CREADOC cuenta con una interfaz sencilla y fácil de usar. En la Figura 5.3 se muestra la pantalla principal.

Del lado derecho se describen las funcionalidades que proporciona el sistema, y del lado izquierdo se encuentra un menú, separado en las secciones: *recuperación*, *clasificación*, *agregar documento*, *inicio* y *contacto*, las cuales se detallan a continuación.

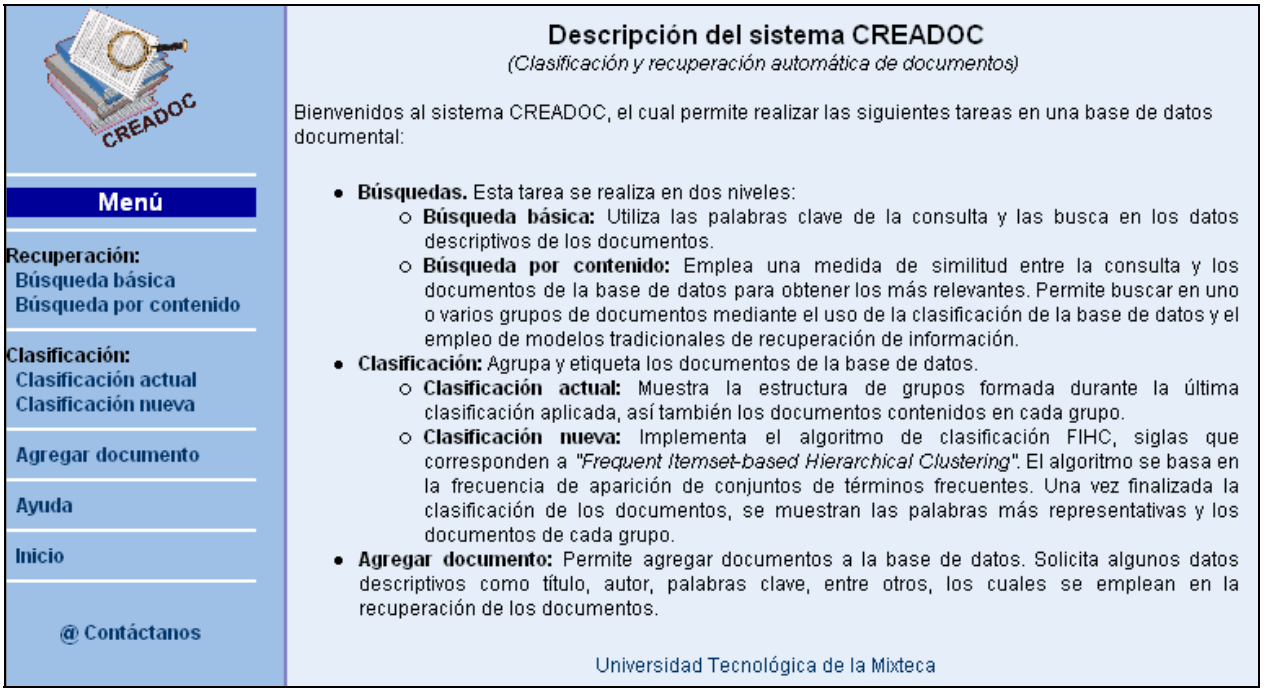


Figura 5.3. Pantalla de inicio de CREADOC.

5.5.1 Interfaz de recuperación

La recuperación de documentos se puede realizar de dos maneras: *búsqueda por metadatos, (*en adelante se denominará *búsqueda básica*) y *búsqueda por contenido*.

a) Búsqueda básica

Este tipo de búsqueda se realiza utilizando los datos descriptivos del documento, los cuales se proporcionan al agregar un documento a la colección. Requiere que el usuario especifique el tipo de búsqueda a realizar, los términos a buscar y que indique si los documentos deben contener todos o sólo algunos de los términos tal como se muestra en la Figura 5.4.

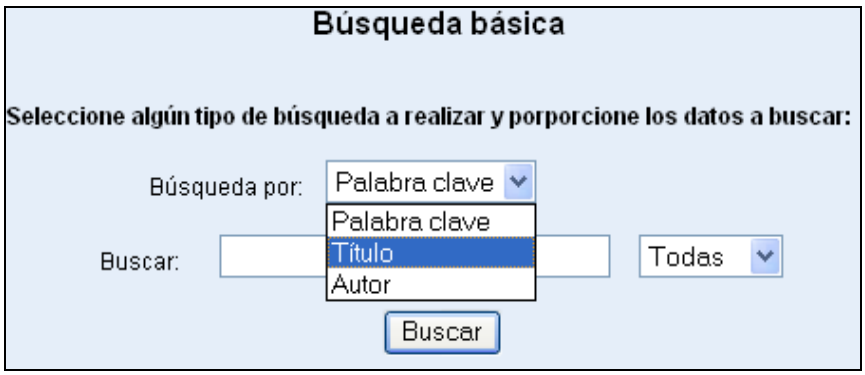


Figura 5.4. Forma para realizar la *búsqueda básica*.

Una vez que se ingresaron los datos y después de presionar el botón *buscar*, el sistema recupera los documentos que contienen los términos de acuerdo al criterio especificado. El resultado del sistema es una lista de documentos ordenados alfabéticamente. Ver la Figura 5.5.

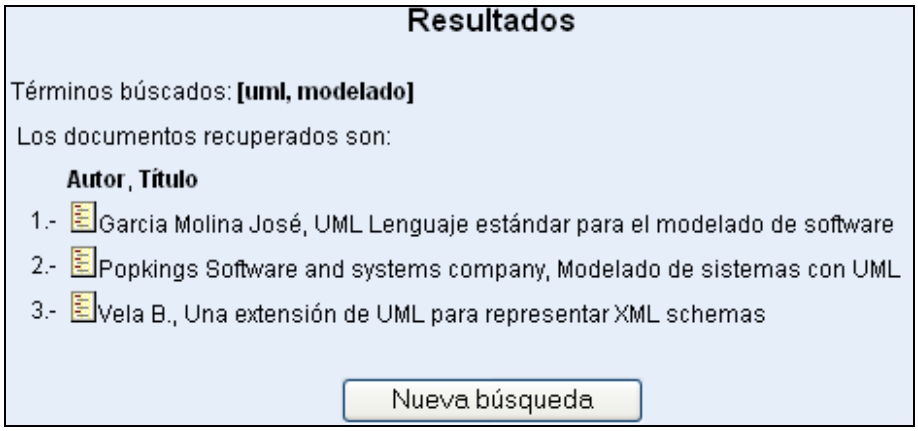


Figura 5.5. Resultados de una *búsqueda básica*.

a) Búsqueda por contenido

La segunda forma de recuperar documentos es mediante una *búsqueda por contenido*, la pantalla para este tipo de consultas se muestra en la Figura 5.6.

En esta consulta es necesario que el usuario especifique los términos a buscar, el porcentaje mínimo de similitud entre la consulta y los documentos y el conjunto de grupos sobre los cuales desea buscar.

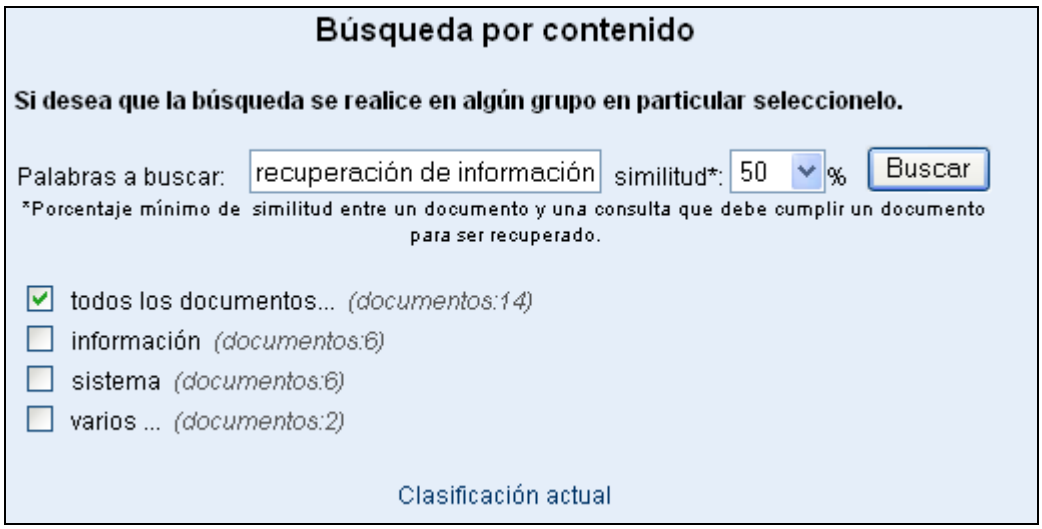


Figura 5.6. Forma de *búsqueda por contenido*.

Cabe señalar que por defecto el sistema busca en el contenido de todos los documentos. Al final de la forma se muestra un enlace para visualizar el conjunto de grupos formados durante la última clasificación. Los resultados se despliegan en forma de lista, ordenada de acuerdo a la similitud del documento con la consulta. La Figura 5.7, muestra un ejemplo de la respuesta de CREADOC a la consulta: *UML, modelado*.

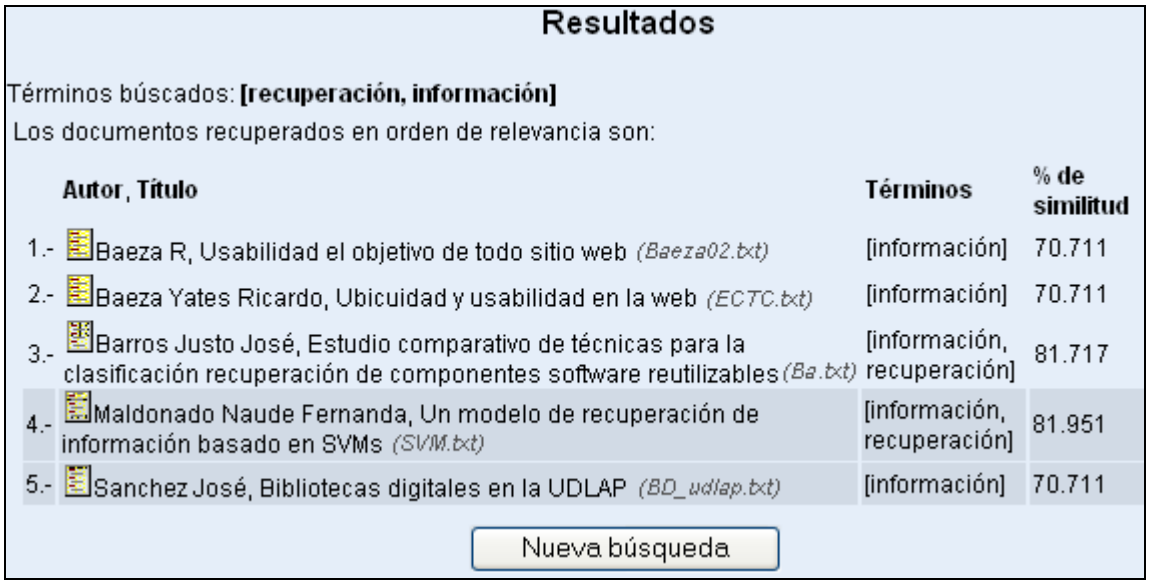


Figura 5.7. Resultados al realizar una *búsqueda por contenido*

Como puede observarse en la lista aparecen al inicio los documentos que tienen mayor similitud con la consulta, (los más relevantes), es decir aquellos documentos que contiene mayor cantidad de términos buscados o cuya frecuencia de aparición de dichos términos es mayor.

5.5.2 Interfaz de clasificación

La sección clasificación consta de dos actividades: *clasificación actual* y *clasificación nueva,* las cuales se detallan a continuación.

a) Clasificación actual

Esta consiste en mostrar la información de los grupos generados durante la última clasificación.

Al inicio de la forma se muestran los parámetros utilizados para generar los grupos, tales como el número total de documentos, el soporte mínimo global, soporte mínimo de grupo y el total de grupos generados. En seguida lista los grupos con sus etiquetas, el número de documentos que contiene el grupo y la información de los documentos. Ver Figura 5.8. Para facilitar la identificación de grupos, estos se separan por una línea.

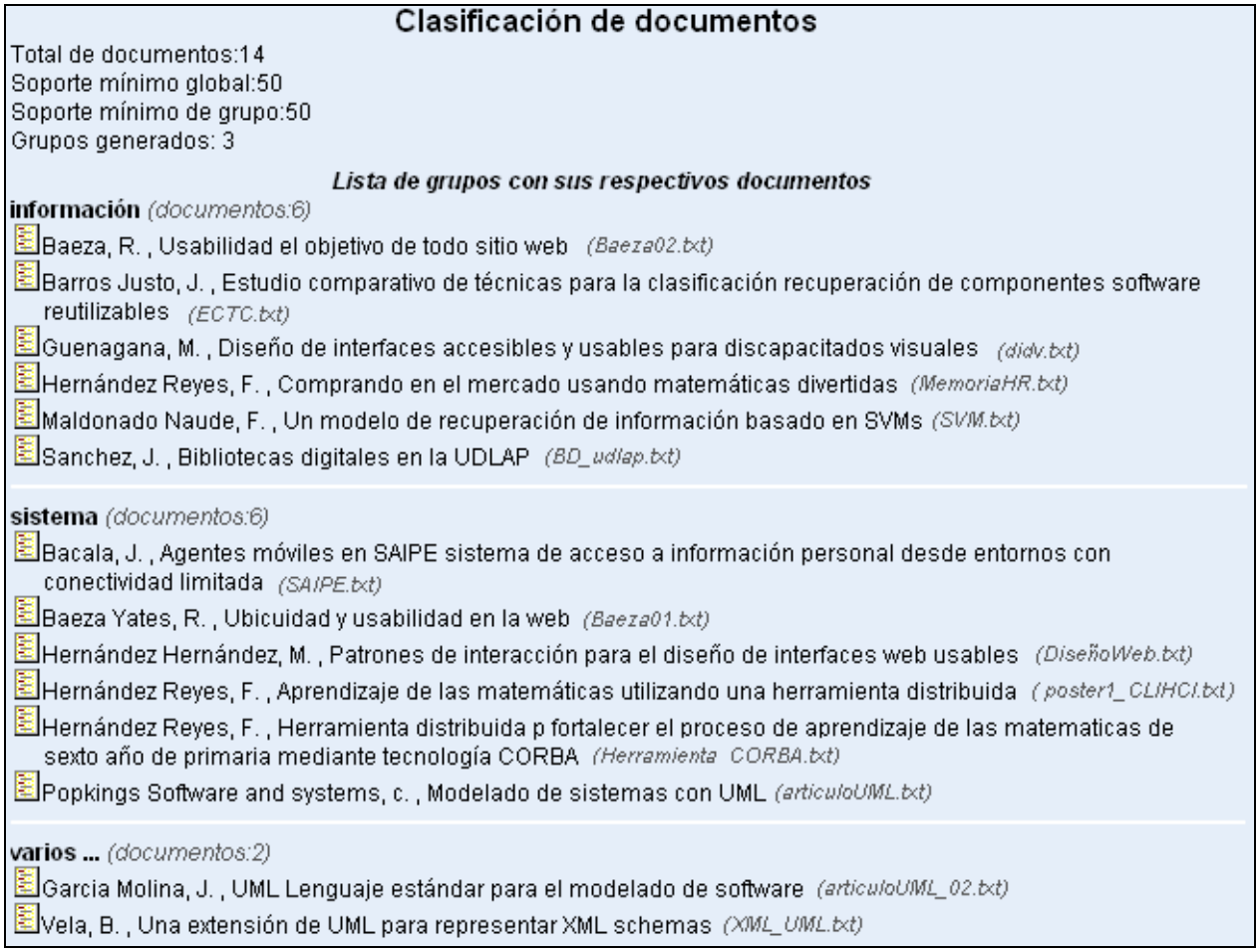


Figura 5.8. Pantalla que muestra la clasificación de los grupos.

El grupo ubicado al final de la lista *(*varios...*)* representa el conjunto de documentos cuya medida de similitud es inferior al valor establecido por el usuario.

b) Clasificación nueva

Para generar una clasificación nueva se requiere que el usuario especifique el *soporte global mínimo* y *soporte grupal mínimo.* La forma en que solicita dichos valores se muestra en la Figura 5.9.

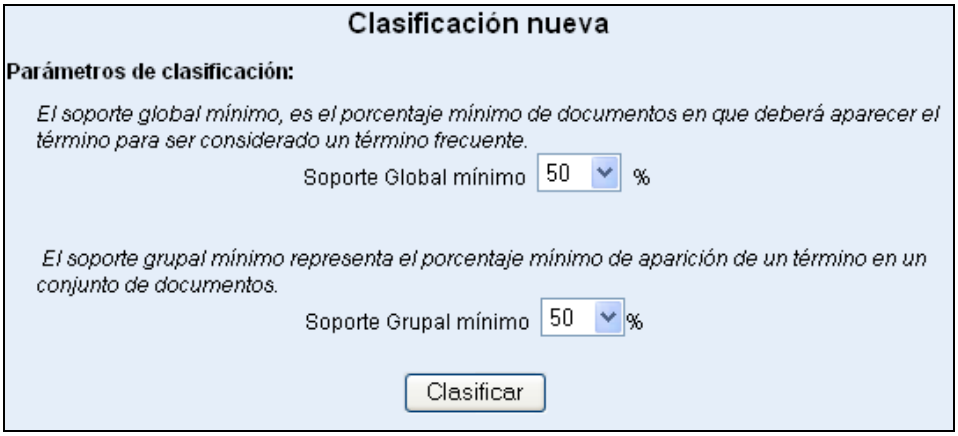


Figura 5.9. Forma para solicitar los parámetros de clasificación.

La clasificación nueva se representa de manera análoga a la clasificación actual. Ver Figura 5.8.

5.5.3 Interfaz de agregar documento

Este módulo permite al usuario agregar un documento a la colección. Para ello se requiere proporcionar los metadatos y su ubicación. Cabe señalar que todos los datos son obligatorios, sin embargo en el caso de los autores se requiere al menos uno; el segundo apellido es opcional. Ver Figura 5.10. Para que el sistema pueda analizar el documento se requiere que el usuario seleccione un documento en formato texto.

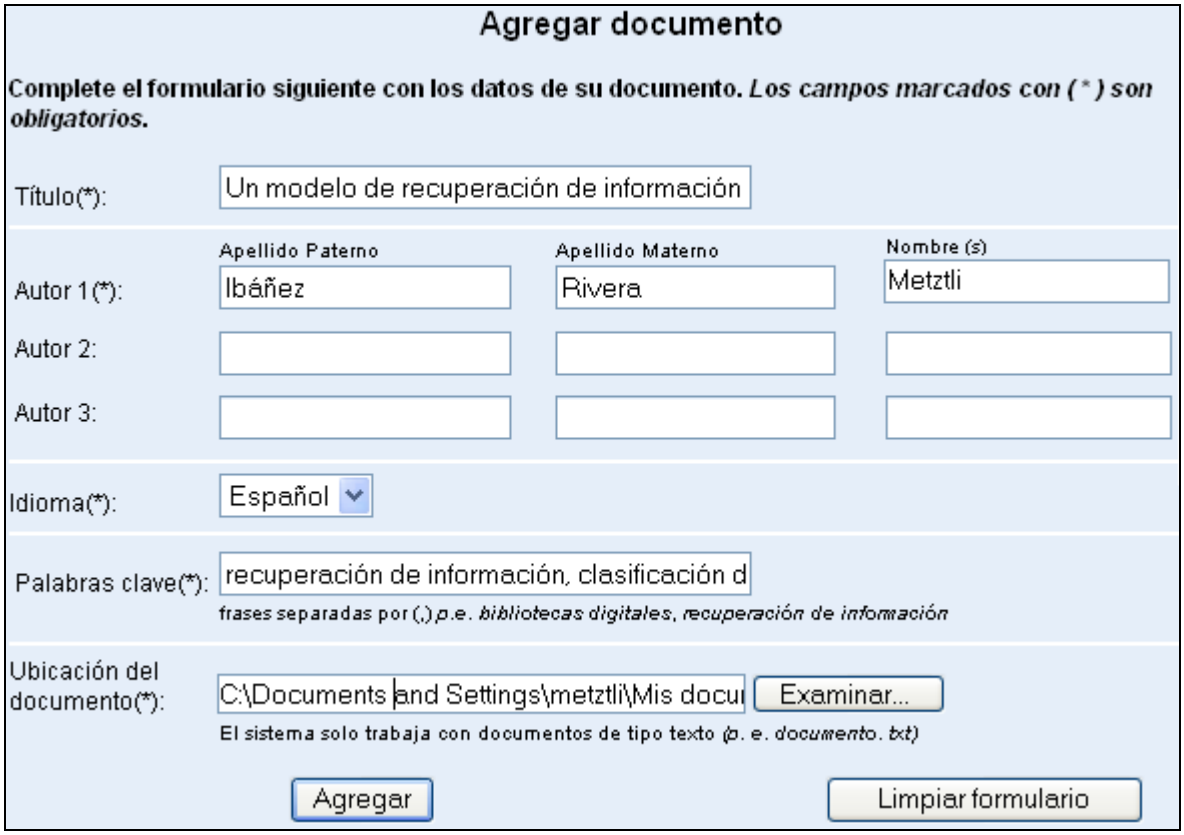


Figura 5.10. Forma para dar de alta un documento.

El parámetro de idioma contempla la opción de idiomas Español e Inglés, dado que al analizar el documento se eliminan las palabras vacías para ambos lenguajes. En el caso de palabras clave, el usuario puede redactarlas en lenguaje natural, el sistema hace un procesamiento de éstas eliminando signos de puntuación y palabras vacías.

Una vez introducidos todos los datos, y presionado el botón *agregar*, CREADOC anexa la información del documento a la colección de documentos y analiza el contenido. Después del análisis del contenido del documento, se muestra al usuario una pantalla confirmando que el documento ha sido agregado para brindar retroalimentación.

5.5.4 Interfaz de contacto

Esta opción se agrega con la finalidad de que el usuario establezca comunicación con los responsables del sistema. Al oprimir este enlace se muestra una forma para enviar un mensaje de correo electrónico en el que el usuario podrá escribir sus dudas, sugerencias o comentarios.

Capítulo 6. Pruebas

Las pruebas tienen como objetivo detectar funcionamientos erróneos del sistema. En el desarrollo de CREADOC se aplicaron diversos tipos de pruebas de usabilidad, de caja negra, caja blanca y complejidad ciclomática. Las pruebas de caja blanca se realizaron durante la implementación del sistema y consistieron en comprobar que se ejecutaran todas las líneas de código y que los procedimientos para llegar a los resultados fueran los correctos. Los tipos de prueba restantes se describen a continuación.

6.1 Pruebas de usabilidad

La usabilidad se considera como la capacidad de eficiencia y satisfacción con la que se comprende, se usa y aprende un software en ciertas situaciones de uso y para ciertos usuarios [ISO 1998].

Para garantizar la facilidad de uso del sistema se aplicaron pruebas de usabilidad a las interfaces del sitio web. Éstas ayudaron a determinar si los contenidos y funcionalidades ofrecidos eran entendibles y de fácil uso para los usuarios [Nielsen 2000].

Las pruebas consistieron en mostrar a un grupo de cinco personas la interfaz del sistema de manera individual. Primero se les pidió realizar ocho actividades, para después cuestionar su percepción sobre dichas actividades. Se contó con la participación de tres observadores y un facilitador. El facilitador fue el guía para los usuarios sobre las tareas a realizar, en tanto, los observadores analizaron las reacciones de los usuarios al momento de llevar a cabo las pruebas para detectar problemas sobre el uso de la interfaz. Todas las pruebas se centraron a los aspectos visuales y organización de contenidos.

Las actividades que se solicitaron a los usuarios son las siguientes:

1. Inserte un documento al sistema
2. Clasifique los documentos solicitando una relevancia de términos frecuentes grupal igual a 60% y frecuencia global igual a 60%
3. Busque los documentos que contienen los términos: interfaz y usuario
4. Busque los documentos que contienen los términos: interfaz y usuario únicamente en los grupos que contienen como nombre alguno de esos términos
5. Busque documentos que en su título contengan las palabras: recuperación de información
6. Busque documentos cuyo autor sea Baeza Yates
7. Busque documentos que contengan como palabra clave el término: UML
8. En las lista de grupos resultado de la clasificación, ¿podría decir que representa las palabras que se encuentran entre las franjas de colores claros?

Al finalizar las tareas se les aplicó un cuestionario de preguntas a los usuarios, la Tabla 6.1 muestra un resumen de las respuestas proporcionadas por los usuarios, de acuerdo a la escala siguiente:

1. Totalmente en desacuerdo
2. Algo en desacuerdo
3. Ni acuerdo ni en desacuerdo
4. De acuerdo en gran parte
5. Totalmente de acuerdo

Tabla 6.1. Respuestas de los usuarios a las pruebas de usabilidad.

Usuario / Preguntas	Usuario1	Usuario2	Usuario3	Usuario4
Interfaz Buena	5	5	4	3
Interfaz clara	4	5	3	3
Interfaz sencilla	4	5	3	3
Tareas fáciles de localizar	4	5	4	4
Mensajes claros	4	5	2	4
Tareas organizadas	5	5	5	4
Colores adecuados	5	5	5	2
Letras legibles	5	4	5	4
Tamaño de letra adecuado	5	4	5	4
Se requiere más imágenes	1	1	2	3

La Figura 6.1 muestra la pantalla principal del prototipo de CREADOC, evaluado durante las pruebas de usabilidad. Para un mejor manejo de espacios se eliminó el encabezado de la página, mostrando la descripción sólo en la página principal. Además, se cambió la distribución del menú, separando las funcionalidades del sistema en recuperación, clasificación, agregar documento e inicio. Ver Figura 6.2.

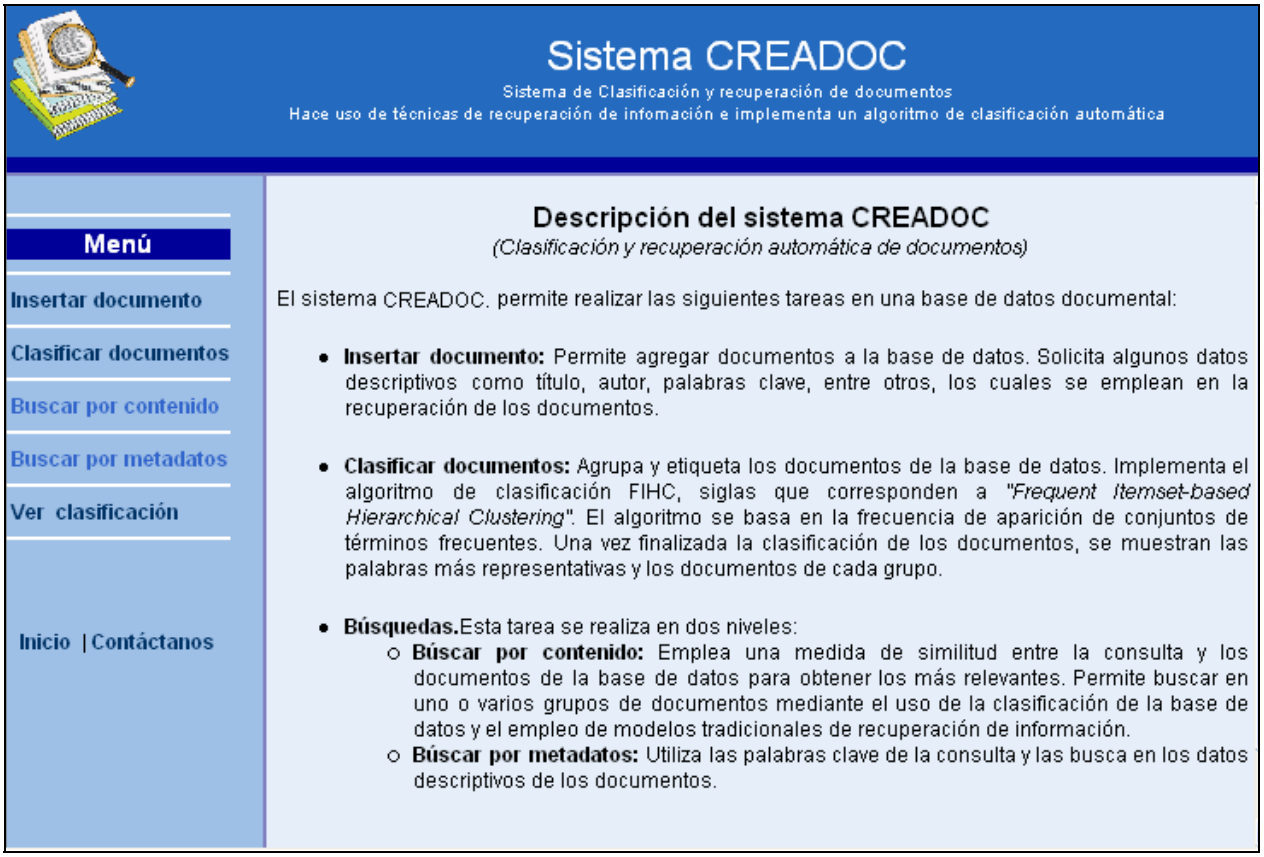


Figura 6.1. Prototipo de la página inicial CREADOC.

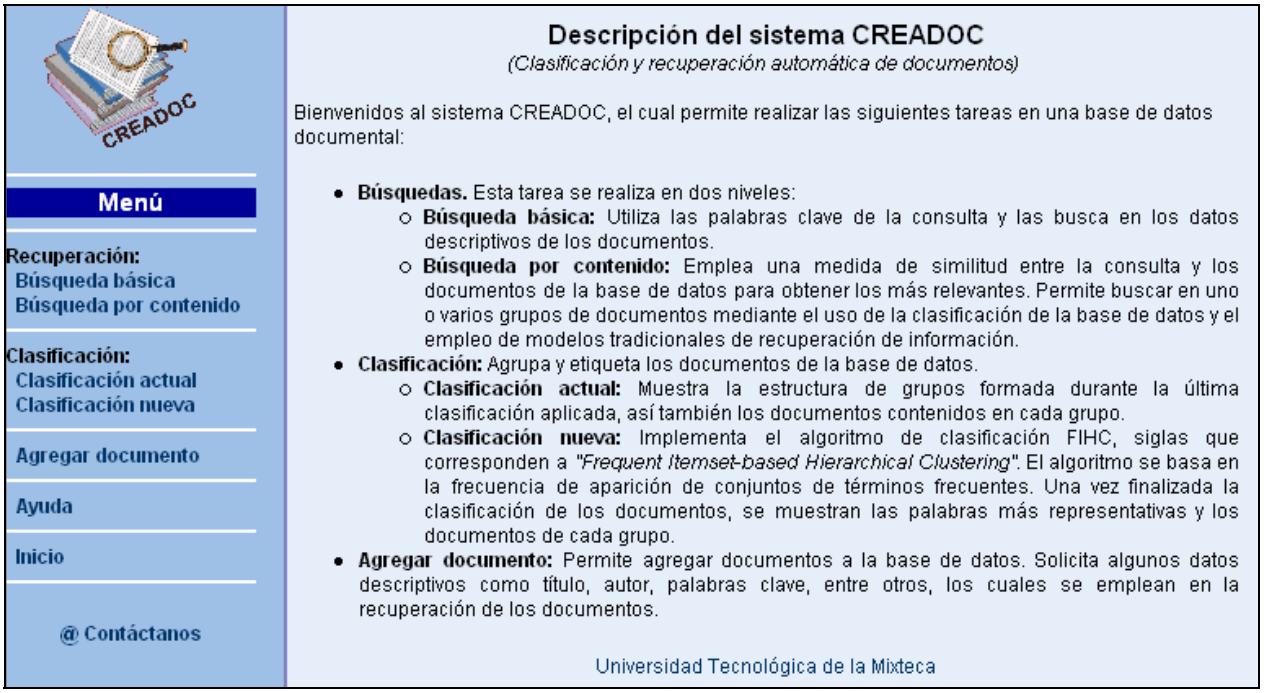


Figura 6.2. Pantalla de inicio de CREADOC después de las pruebas de usabilidad.

El prototipo de la *búsqueda por metadatos* resultó ser bastante confusa para los usuarios, (ver Figura 6.3), por lo que se reemplazó el título *Búsqueda por metadatos* por *Búsqueda básica*. Además, se rediseñó la forma de solicitar los parámetros de búsqueda. Ver Figura 6.4.

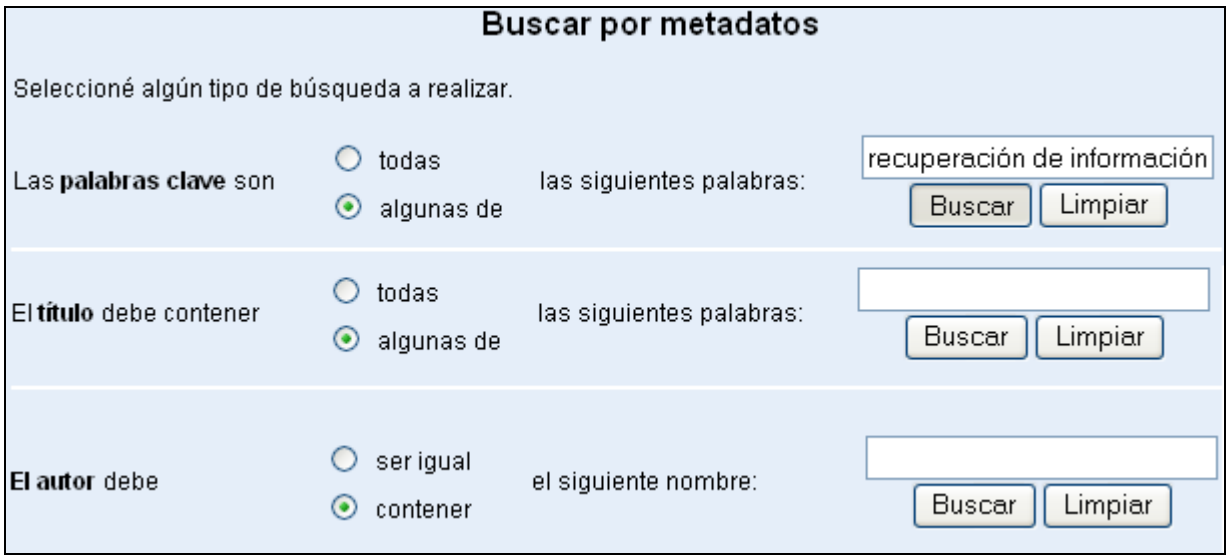


Figura 6.3. Prototipo para la *búsqueda por metadatos*.

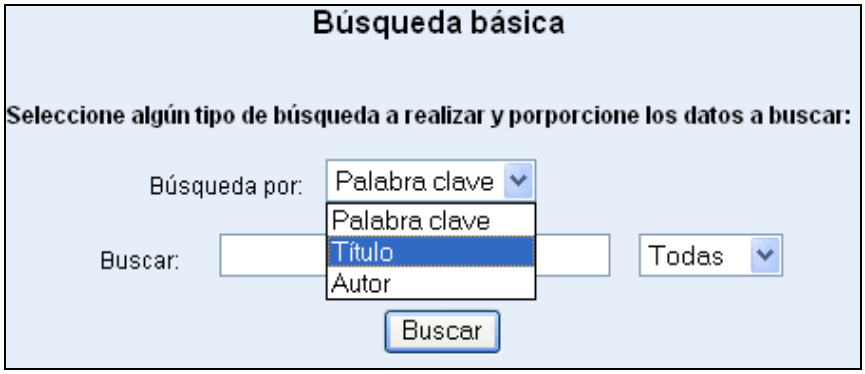


Figura 6.4. Forma para realizar la *búsqueda básica* después de las pruebas de usabilidad.

En la interfaz para clasificar, se observó que los usuarios no leían la información que se encontraba como pie de página, (ver Figura 6.5), por lo que se reemplazaron por comentarios, colocándolos antes de la selección de valores. Ver Figura 6.6.

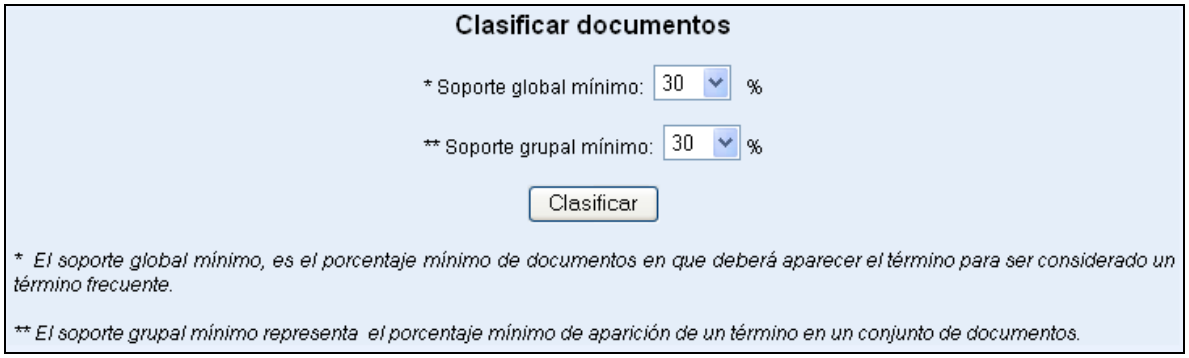


Figura 6.5. Prototipo de la interfaz para *clasificar documentos*.

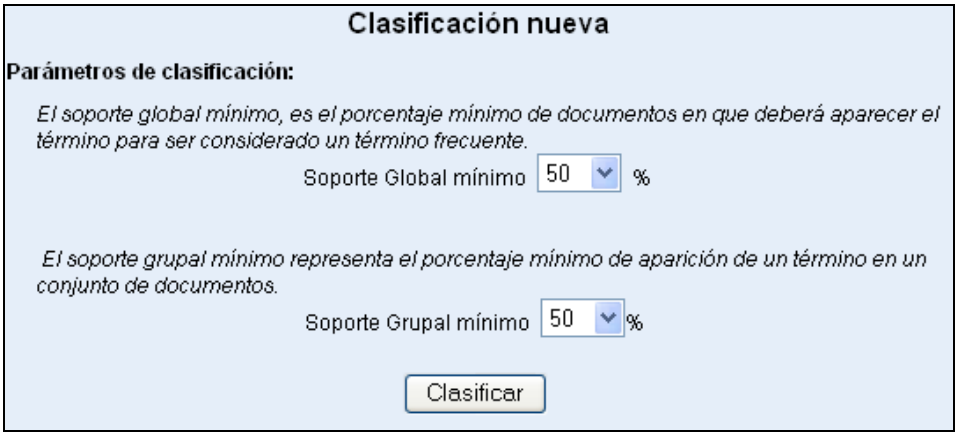


Figura 6.6. Forma para solicitar los parámetros de clasificación después de las pruebas de usabilidad.

De acuerdo a lo observado, se optó por colocar un vínculo para ver la información de los grupos al realizar búsquedas por contenido, dicha información se muestra en una ventana emergente. Se agregaron botones para realizar una nueva búsqueda en las pantallas de resultados de búsquedas. También se agregaron instrucciones a las actividades que permite realizar el sistema y se ejemplificó sobre la manera de cómo introducir los términos a buscar.

Por último se rediseñó el logotipo del sistema y se implementó un módulo de ayuda, el cual se muestra en una ventana emergente.

Los resultados de las pruebas de usabilidad permitieron adecuar los elementos de la interfaz, con el fin de atender los requerimientos de los usuarios y satisfacer sus expectativas. Una sugerencia que no se incorporó fue la visualización de los documentos, debido a que no fue un objetivo de la tesis y a que CREADOC no almacena el documento, sólo analiza su contenido y obtiene una estadística de los términos relevantes. Esto reduce el espacio requerido y evita conflicto con los derechos de autor.

6.2 Pruebas de caja negra

Este tipo de pruebas se utiliza para evaluar el funcionamiento correcto del sistema. Se evaluaron las dos funcionalidades principales: la clasificación y la recuperación de documentos. En la clasificación se consideró el tiempo de respuesta y la similitud de los grupos generados por CREADOC contra una clasificación de expertos. Al evaluar la recuperación se consideraron parámetros que involucraran la cantidad de documentos recuperados y documentos relevantes a la consulta.

La Tabla 6.2 muestra un resumen de las características de los documentos utilizados para este tipo de pruebas.

Tabla 6.2. Documentos utilizados durante las pruebas del sistema.

No. Doc	Título	Número de palabras	Tamaño en kilobytes
1.	Agentes móviles en SAIPE con conectividad limitada	2890	19
2.	Aprendizaje de las matemáticas utilizando una herramienta distribuida	1831	12
3.	Comprando en el mercado usando matemáticas divertidas	3444	23
4.	Diseño de interfaces accesibles y usables para discapacitados visuales	3342	22
5.	Estudio comparativo de técnicas para la clasificación recuperación de Componentes software reutilizables	4727	33
6.	Herramienta distribuida para fortalecer el proceso de aprendizaje de las matemáticas mediante tecnología CORBA	4288	28
7.	Modelado de sistemas con UML	6914	47
8.	Patrones de interacción para el diseño de interfaces web usables	1913	13
9.	Ubicuidad y usabilidad en la web	6924	43
10.	U-dl-a Bibliotecas digitales en la udla	4940	34
11.	UML Lenguaje estándar para el modelado de software	1483	9
12.	Un modelo de recuperación de información basado en SVMs	1939	13
13.	Una extensión de UML para representar XML Schemas	1913	13
14.	Usabilidad: el objetivo de todo sitio web	2652	16

Como puede observarse, el tamaño y número de palabras de los archivos no es homogéneo. Para la muestra, en promedio el tamaño de un archivo es de 23 Kb y está formado por 3514 palabras. Observe que el resultado de las pruebas dependerá de estos datos.

6.2.1 Evaluación de la clasificación de documentos

Para evaluar la clasificación de documentos se realizaron varias pruebas. La primera consistió en clasificar varios grupos de documentos para visualizar los tiempos consumidos por la clasificación, iniciando con tres documentos hasta un máximo de 14. En la segunda prueba, se evaluó la efectividad del algoritmo FIHC para generar grupos. Se generaron varias clasificaciones las cuales se compararon con una clasificación de expertos.

a) Tiempo de respuesta del sistema al clasificar

En esta primera prueba, se analizaron los tiempos que el sistema requirió para clasificar el conjunto de documentos, aplicando distintos valores para el soporte global (*global support, GS*) y soporte de grupo (*cluster support, CS*). La Tabla 6.3 muestra algunos de los resultados arrojados por el sistema al clasificar varios conjuntos de documentos con diferentes valores de soporte global y grupal.

La interpretación de la primera muestra se hace de la siguiente manera: con 10 documentos en el sistema y solicitando un soporte global de 80% y de grupo de 75%, el sistema requirió de 0.282 segundos para generar la clasificación. Cabe señalar que la medición del tiempo se hizo mediante una clase que lee el reloj del sistema operativo antes y al terminar la clasificación.

Tabla 6.3. Desempeño de la clasificación en segundos.

Número de documentos	GS=80% CS=75%	GS=50% CS=50%	GS=30% CS=30%	GS=20% CS=20%	GS=10% CS=10%
10	0.282	0.437	1	6.329	2.95
11	0.141	0.329	0.641	1.172	3.843
12	0.172	0.343	0.625	1.032	3.89
13	0.203	0.359	0.64	1.125	3.75
14	0.156	0.344	0.547	1.11	4.953

El desempeño de FIHC depende de los valores que se especifiquen para sus soportes, así como el número de términos frecuentes de los documentos. Como puede observarse, al incrementar la cantidad de términos frecuentes, el tiempo para clasificar también aumenta. Sucede lo mismo al disminuir el soporte global. Esto se debe a que el algoritmo a priori realiza un mayor número de empates entre términos.

b) Evaluación de FIHC

La evaluación de la precisión de FIHC se resume en las Tablas 6.4, 6.5 y 6.6. La primera muestra la clasificación de expertos, el encabezado contiene los términos que el experto consideró describen al conjunto de documentos y en seguida se muestran los títulos de los documentos.

Tabla 6.4. Clasificación de expertos.

Clasificación de expertos
• agentes móviles, usuario • Agentes móviles en SAIPE con conectividad limitada • interfaces, diseño web, usabilidad, usuario • Diseño de interfaces accesibles y usables para discapacitados visuales • Patrones de Interacción para el Diseño de Interfaces web usables • Ubicuidad y Usabilidad en la web • Usabilidad El objetivo de todo sitio web • matemáticas, aprendizaje, fracciones • Aprendizaje de las Matemáticas Utilizando una Herramienta Distribuida • Comprando en el Mercado usando Matemáticas Divertidas • Herramienta distribuida para fortalecer el aprendizaje de las matemáticas mediante tecnología CORBA • modelado, uml • Modelado de Sistemas con UML • UML Lenguaje estándar para el modelado de software • Una extensión de UML para representar XML Schemas • recuperación de información, documentos • Estudio comparativo de técnicas para la clasificación recuperación de componentes de software reutilizables • U-dl-a bibliotecas digitales en la udla • Un modelo de recuperación de información basado en SVMs

La representación jerárquica de la clasificación propuesta por expertos se muestra en la Figura 6.7.

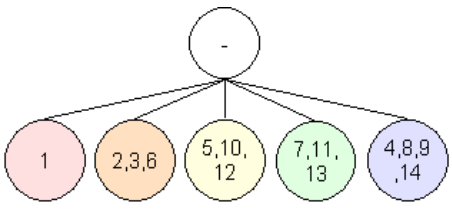


Figura 6.7. Clasificación de expertos.

Una vez obtenida la clasificación humana, se generaron clasificaciones en CREADOC, variando los soportes globales y de grupo.

Para generar la clasificación de la Tabla 6.4, se utilizaron soportes del 20 y 60%, lo cual indica que para que un término fuera considerado término frecuente debió aparecer en el 20% de los documentos y para que un conjunto de documentos formaran un grupo, los documentos debían compartir un 60% de los mismos términos frecuentes. El sistema tardó 0.813 segundos en crear los grupos.

Tabla 6.5. Clasificación con GS=20% y CS=60%.

Clasificación con GS=20% y CS=60%
actividades, aprendizaje, enseñanza, fracciones, uso *(documentos:3)* • Aprendizaje de las matemáticas utilizando una herramienta distribuida • Comprando en el mercado usando matemáticas divertidas • Herramienta distribuida para fortalecer el aprendizaje de las matemáticas mediante tecnología CORBA **información, desarrollo** *(documentos:1)* • Estudio comparativo de técnicas para la clasificación recuperación de componentes software reutilizables **información, documentos** *(documentos:1)* • Un modelo de recuperación de información basado en SVMs **información, Interfaz, sistema** *(documentos:1)* • Agentes móviles en SAIPE con conectividad limitada **uml** (documentos:3) • UML Lenguaje estándar para el modelado de software • Modelado de Sistemas con UML • Una extensión de UML para representar XML Schemas **usuario** *(documentos:1)* • U-dl-a bibliotecas digitales en la udla **usuario, información, sistema** *(documentos:2)* • Ubicuidad y Usabilidad en la web • Patrones de Interacción para el diseño de interfaces web usables **usuario, información, contenido, diseño, usabilidad, usuarios** *(documentos:2)* • Usabilidad El objetivo de todo sitio web • Diseño de interfaces accesibles y usables para discapacitados visuales

Esta clasificación generó ocho grupos, en lugar de los cinco propuestos por los expertos. Seis documentos quedaron en los mismos grupos, un total de 12 documentos fueron descritos por alguno de los términos propuestos. Ver Figura 6.8. Los grupos vacíos no se contabilizan.

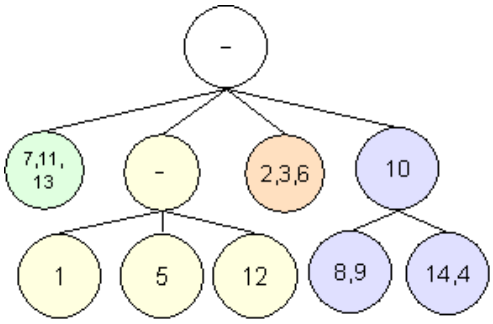


Figura 6.8. Clasificación con GS=20% y CS=60%.

Para la siguiente clasificación se utilizó un soporte global de 10% y grupal de 60%. Ver Tabla 6.6. Esta clasificación generó 9 grupos de documentos en 4 segundos. Se sigue teniendo una similitud con la clasificación propuesta, siendo más clara en los grupos *UML, fracciones, usabilidad*.

Tabla 6.6. Clasificación con GS=10% y CS=60%.

Clasificación CREADOC con GS=10% y CS=60%
actividades, aprendizaje, enseñanza, fracciones, uso *(documentos:3)* • Aprendizaje de las matemáticas utilizando una herramienta distribuida • Comprando en el mercado usando matemáticas divertidas • Herramienta distribuida para el aprendizaje de las matemáticas mediante tecnología CORBA
información, clasificación, conjunto, recuperación *(documentos:1)* • Un modelo de recuperación de información basado en SVMs
información, desarrollo, sistema *(documentos:1)* • Estudio comparativo de técnicas para la clasificación recuperación de componentes software reutilizables
información, usuario, diseño, contenido, usabilidad, usuarios *(documentos:2)* • Usabilidad El objetivo de todo sitio web • Diseño de interfaces accesibles y usables para discapacitados visuales
información, usuario, sistema, forma, usabilidad *(documentos:1)* • Ubicuidad y Usabilidad en la web
información, usuario, sistema, interfaz *(documentos:2)* • Agentes móviles en SAIPE con conectividad limitada • Patrones de Interacción para el Diseño de Interfaces web usables
Información, usuario, documentos *(documentos:1)* • U-dl-a bibliotecas digitales en la UDLA
uml, clase *(documentos:2)* • Modelado de Sistemas con UML • Una extensión de UML para representar XML Schemas
uml, diseño, modelado *(documentos:1)* • UML Lenguaje estándar para el modelado de software

En la Figura 6.9 se ilustra la agrupación de los documentos. Sin considerar las etiquetas de los grupos.

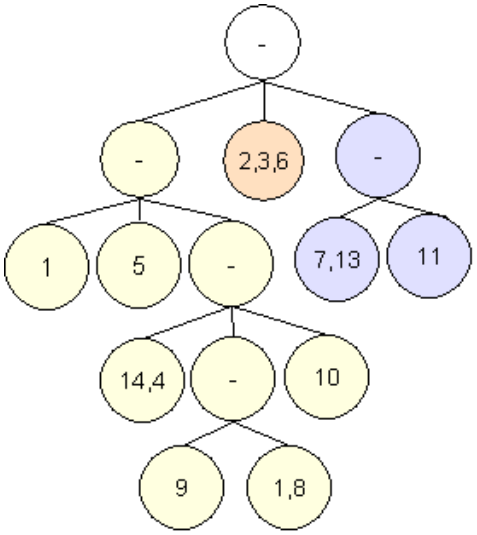


Figura 6.9. Clasificación con GS=10% y CS=60%.

Por último se generó una nueva clasificación con soporte global de 25% y grupal de 65%. CREADOC generó una clasificación más general que la propuesta por los expertos, compuesta por 6 grupos y un grupo denominado varios, el cual contiene los documentos que no compartieron la similitud mínima solicitada por el usuario para formar o pertenecer a un grupo, dicha clasificación se realizó en 0.656 segundos. Ver Tabla 6.7.

Tabla 6.7. Clasificación con GS=25% y CS=65%.

Clasificación con GS=25% y CS=65%
desarrollo *(documentos:3)* • Estudio comparativo de técnicas para la clasificación recuperación de componentes software reutilizables • UML Lenguaje estándar para el modelado de software • Comprando en el Mercado usando Matemáticas Divertidas **diseño, información, usabilidad, usuario** *(documentos:3)* • Usabilidad: el objetivo de todo sitio web • Diseño de interfaces accesibles y usables para discapacitados visuales • Un modelo de recuperación de información basado en SVMs **sistema, interfaz** *(documentos:3)* • Agentes móviles en SAIPE con conectividad limitada • Aprendizaje de las matemáticas utilizando una herramienta distribuida • Herramienta distribuida para fortalecer el aprendizaje de las matemáticas mediante tecnología CORBA **sistema, uso** *(documentos:1)* • Modelado de Sistemas com UML **sistema, usuario** *(documentos:2)* • Ubicuidad y Usabilidad en la web • Patrones de interacción para el diseño de interfaces web usables **usuario** *(documentos:1)* • U-dl-a bibliotecas digitales en la udla **varios ...** *(documentos:1)* • Una extensión de UML para representar XML Schemas

Esta clasificación comparte una menor similitud con la clasificación propuesta por el usuario. Como puede observarse en la Figura 6.10.

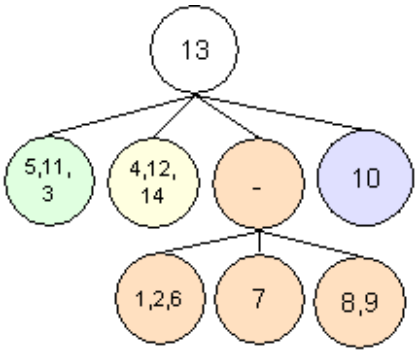


Figura 6.10. Clasificación con GS=25% y CS=65%.

CREADOC permite al usuario generar diferentes clasificaciones, especificando el valor mínimo de similitud para considerar aun término frecuente (frecuencia global) y el mínimo de similitud entre documentos para formar un grupo (frecuencia grupal). Lo cual facilita al usuario generar varias clasificaciones y dejar la que se considere que se apega más su percepción como experto.

6.2.2 Evaluación de recuperación de información

Las pruebas de recuperación de información se llevaron a cabo en dos etapas, la primera consistió en evaluar la efectividad del sistema en la *búsqueda por contenido*, implementa el modelo de espacios vectoriales. La segunda evaluó la *búsqueda básica*, la cual consiste en realizar una consulta booleana a la base de datos.

Para evaluar la eficiencia de los métodos de recuperación de información se consideran dos parámetros principales: *precisión* y *recall*. Estos parámetros evalúan el conjunto de documentos recuperados y los documentos no recuperados, tomando en cuenta el conjunto de documentos relevantes e irrelevantes para una consulta [Baeza & Ribeiro 1999].

Precisión es la fracción de documentos que han sido recuperados y que es relevante, definida como:

$$precisión = \frac{Documentos\ Relevantes\ Recuperados}{Total\ de\ Documento\ Recuperados} \times 100$$

Recall es la fracción de documentos relevantes que han sido recuperados, se define de la siguiente manera:

$$recall = \frac{Documentos\ Relevantes\ Recuperados}{Documento\ Relevantes} \times 100$$

Para realizar estas pruebas fue necesario que el experto conociera el contenido de los documentos y en base a su experiencia, realizara consultas y pronosticara la respuesta correcta. La Tabla 6.8 muestra en letras negritas la consulta realizada y en seguida lista el conjunto de documentos que satisfacen su consulta.

Tabla 6.8. Consultas y resultados de expertos

Consultas y resultados de expertos
recuperación de información
• Estudio comparativo de técnicas para clasificación recuperación de componentes
• Un modelo de recuperación de información basado en SVMs
• U-dl-a bibliotecas digitales en la udla
modelado de software
• Modelado de Sistemas con UML
• UML Lenguaje estándar para el modelado de software
• Una extensión de UML para representar XML Schemas
• Comprando en el Mercado usando Matemáticas Divertidas

a) Búsqueda por contenido

La evaluación de la *búsqueda por contenido* consistió en comparar los documentos recuperados por el sistema con los propuestos por los expertos para obtener los valores de precisión y recall. Esta búsqueda contempla los términos relevantes del documento junto con las palabras clave las cuales se introducen al agregar un documento a la colección.

Cabe señalar que para realizar las consultas mostradas a continuación se solicitó al sistema una similitud mínima de 50%, es decir, el documento debía contener por lo menos la mitad de los términos solicitados para ser recuperado. La *búsqueda por contenido*, se realizó en todo el conjunto de documentos, no se seleccionó algún grupo en particular.

La primera consulta realizada fue *recuperación de información*. La respuesta del sistema se muestra en la Tabla 6.9, la cual consistió en cinco documentos, ordenados de acuerdo al valor de similitud. Los dos primeros documentos de la lista corresponden a los propuestos por los expertos.

Tabla 6.9. Resultados de la consulta 1

Resultados de la consulta 1		
Título del documento	**Términos**	**Similitud (%)**
• Estudio comparativo de técnicas para la clasificación recuperación de componentes software reutilizables	[información, recuperación]	66,722
• Un modelo de recuperación de información basado en SVMs	[información, recuperación]	66,913
• U-dl-a bibliotecas digitales en la UDLA	[información]	57,735
• Usabilidad El objetivo de todo sitio web	[información]	57,735
• Ubicuidad y Usabilidad en la web	[información]	57,735

La precisión del sistema para esta consulta fue de 60% ya que de los cinco documentos recuperados tres eran relevantes para la consulta del usuario. El *recall* fue de 100%, esto es, recuperó todos los documentos relevantes existentes. En la comunidad de RI, los porcentajes de *recall* y precisión superiores al 60% se consideran apropiados [Baeza 1999].

Los resultados para la segunda consulta: *modelado de software* se muestran en la Tabla 6.10. El sistema recuperó 5 documentos.

Tabla 6.10. Resultados de la consulta 2

Resultados de la consulta 2		
Título del documento	**Términos**	**Similitud (%)**
• UML Lenguaje estándar para el modelado de software	[software, modelado]	80,497
• Estudio comparativo de técnicas para la clasificación recuperación de componentes software reutilizables	[software]	57,735
• Patrones de Interacción para el Diseño de Interfaces web usables	[software]	57,735
• Comprando en el Mercado usando Matemáticas Divertidas	[software]	57,735
• Modelado de Sistemas con UML	[modelado]	57,735

La precisión del sistema fue del 60%, ya que de los cinco documentos que recuperó tres documentos son relevantes. En tanto su r*ecall* fue de 75%, (recuperó tres documentos relevantes de los cuatro que se esperaban).

b) Búsqueda básica

La evaluación de la *búsqueda básica* consistió en ver que el sistema recuperara los documentos que contenían todos o algunos de los términos solicitados por el usuario. Esta búsqueda contempla sólo los descriptores del documento, los cuales son especificados por el usuario al momento de registrar un documento.

En la primera prueba se solicitaron los documentos que tuvieran como palabra clave alguno de los términos: *recuperación de información*. Los documentos resultantes se muestran en la Tabla 6.11.

Tabla 6.11. Resultados de la consulta 3.

Resultados de la consulta 3
• Estudio comparativo de técnicas para la clasificación recuperación de componentes software reutilizables • UML Lenguaje estándar para el modelado de software • Aprendizaje de las Matemáticas Utilizando una Herramienta Distribuida • Un modelo de recuperación de información basado en SVMs • Modelado de Sistemas con UML • U-dl-a bibliotecas digitales en la UDLA

La precisión del sistema para esta consulta fue de 50% ya que de los seis documentos recuperados tres eran relevantes para la consulta del usuario. El *recall* fue de 100%, esto es, recuperó todos los documentos relevantes existentes.

En la siguiente prueba se solicitó al sistema recuperar los documentos con título: *modelado de software*. La respuesta del sistema se muestra en la Tabla 6.12.

Tabla 6.12. Resultados de la consulta 4

Resultados de la consulta 4
• UML lenguaje estándar para el modelado de software

La precisión del sistema fue del 100% porque recuperó solo un documento y éste fue relevante. En tanto, el *recall* fue de 25%, (recuperó un documento relevante de los cuatro esperados).

La figura 6.11 muestra la precisión y recall observados en las pruebas de recuperación de información para el modelo booleano (búsqueda básica) y para el modelo de espacios vectoriales (búsqueda por contenido).

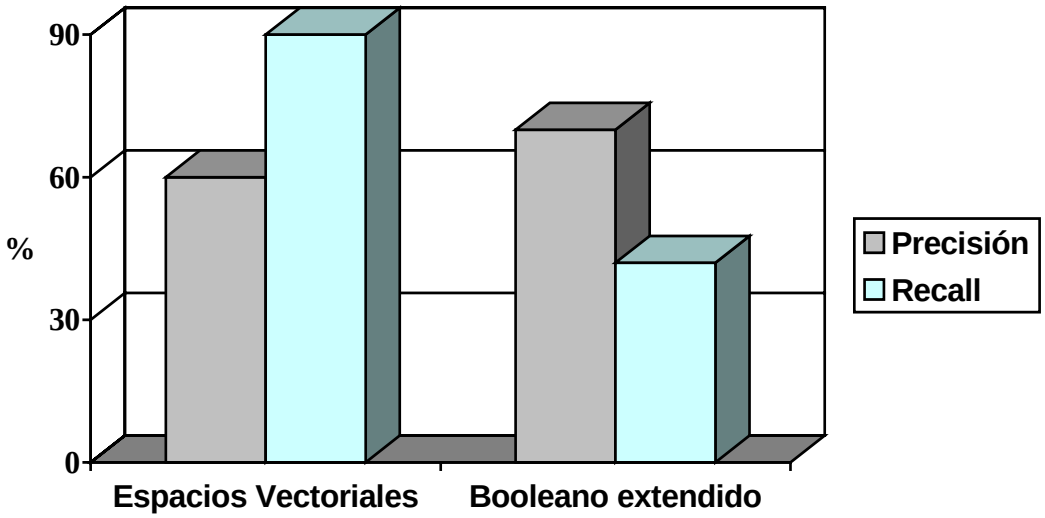


Figura 6.11. Precisión y recall de la búsqueda básica y búsqueda por contenido .

Capítulo 7. Conclusiones y trabajo futuro

Este documento presenta el diseño e implementación del sistema CREADOC, el cual hace uso de un método de clasificación automática basado en conjuntos de términos frecuentes. A su vez, incorpora un modelo de recuperación de información para encontrar documentos relevantes. CREADOC analiza los documentos proporcionados por el usuario y almacena en la base de datos sólo un grupo de términos descriptivos, lo cual reduce la cantidad de espacio requerido para cada documento.

En colecciones extensas de documentos, la recuperación es una tarea lenta. Sin embargo, la aplicación de algoritmos de clasificación acelera el proceso, dado que la búsqueda se realiza sólo en los grupos seleccionados por el usuario y no en toda la colección de documentos. Además, de acuerdo a las pruebas, se obtuvo un aumento en la precisión de los resultados. En la operación del sistema, se espera que la clasificación no se realice tan frecuentemente como la recuperación de información.

Sobre el funcionamiento del algoritmo FIHC, se observó que entre más bajo sea el soporte global, existirán mayor número de términos que describan a los documentos y por lo tanto, se formarán grupos más específicos.

Por otro lado, el procesamiento de las consultas permite que la extracción de palabras clave se realice de forma automática. Es importante señalar que es irrelevante el orden de las palabras clave. Debido a que las consultas se introducen en lenguaje natural, los usuarios no necesitan expresarlas de acuerdo a alguna sintaxis particular.

En comparación con la búsqueda booleana, en la *búsqueda por contenido* con el modelo de espacios vectoriales, existe el inconveniente de que al realizar una búsqueda de un término que aparece en todos los documentos no se recupera ninguno, a menos que se considere un valor muy pequeño de similitud.

Algunos sistemas usan técnicas de lematización, la cual permite extraer prefijos comunes de forma que un conjunto de palabras con la misma raíz se trata como una sola. La lematización también se emplea para reducir el espacio de almacenamiento de los documentos o para representar familias de palabras [Korenius et al. 2004]. Esta tesis hizo uso del algoritmo de Porter [Porter 1980], con el cual se redujo el número de palabras clave almacenadas, aunque disminuyó la precisión de la recuperación. Esta situación se detectó también al evaluar la clasificación con y sin lematización, en donde se observó que se crean más grupos de documentos con pocos términos en común cuando se lematiza, por lo que se descartó la aplicación de este algoritmo de en la versión final del sistema.

CREADOC es un sistema que puede cambiar su interfaz con facilidad dado que las características de configuración se concentran en un solo archivo. Su implementación con Servlets facilita las modificaciones en la interfaz sin afectar la funcionalidad del mismo o requerir de algún proceso de compilación.

Por otro lado, la funcionalidad de CREADOC puede extenderse como sigue:

1. *Empleo de otros formatos de documentos*.- El sistema trabaja únicamente con documentos en formato texto, sin embargo, pueden ampliarse los tipos de archivos a analizar. Para analizar archivos en otro formato, se requerirá un módulo adicional para abrir el documento y procesarlo con el propósito de extraer los términos descriptivos

2. *Visualización del contenido de documentos*.- En los resultados de una búsqueda, el sistema muestra la información descriptiva de los documentos relevante. Se puede crear un módulo para acceder al documento directamente. Actualmente, la versión final de CREADOC no requiere almacenar los documentos en la base de datos o conocer su ubicación

3. *Incorporación de tesaurus o diccionario de sinónimos*.- Está es una herramienta que sería de mucha utilidad durante la recuperación de documentos, debido a que el sistema automáticamente realizaría la consulta no sólo de los términos buscados, sino también de palabras que contengan conceptos similares, por lo que se podría anexar un módulo que expanda la consulta

4. *Coincidencia aproximada a caracteres*.- La incorporación de esta técnica permitiría recuperar documentos que contienen palabras lexicográficamente similares, es decir, extender la recuperación de información actual a una capaz de manejar errores

Referencias

- Agrawal R. Ramakrishnan S. 1994. *Fast algorithms for mining association rules in large databases*. In VLDB-94, September.

- Baeza Yates R., Ribeiro-Neto B. 1999. *Modern information retrieval*. Berthier. Addison-Wesley.

- Brücher, H., Knolmayer, G., Mittermayer, M. 2002. Document classification methods for organizing explicit knowledge. En *Proceedings of the Third European Conference on Organizational Knowledge, Learning, and Capabilities 2002* (Athen, 2002).

- Cimiano, P., Hotho, A., Staab S. 2004. Comparing conceptual, partitional and agglomerative clustering for learning taxonomies from text. En *Proceedings of the European Conference on Artificial Intelligence* (Proc of ECAI-2004, Agosto), Valencia, España.

- Florian B. Martin E. Xiaowei X. 2002. Frequent term-based text clustering. En *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, Alberta, Canada. 436-442

- Fung, B., Wang, K., & Ester, M. 2003 Hierarchical document clustering using itemsets. En *Proc of the 3rd SIAM International Conference on Data Mining* (SDM'03, Mayo). San Francisco, CA, United States 59-70.

- Grossman D. A. Frieder O. 1998. *Information Retrieval: Algorithms and Heuristics*. Kruler Academic Publishers, U. S. A.

- International Standard 1998. *ISO 9241-11:1998. Ergonomic requirements for office work with visual display terminals (VDTs)-Part 11: Guidance on usability.*

- Jain A. K., Murty M. N., Flynn P. J. 1999. *Data clustering: a review*. ACM Computing Surveys (CSUR) Volume 31 Issue 3: 264-323.

- Jain A. K., Dubies R. C. 1998. *Algorithms for clustering data*. Prentice Hall, Inc., Upper Saddle River, NJ.

- Jiří, H. 2002. Document classification in a digital library. Reporte técnico DCSE/TR-2002-04. Department of Computer Science and Engineering, University of West Bohemia in Pilsen, Czech Republic., Abril.

- Korenius T., Laurikkala J., Järvelin K., Juhola M. 2004. Stemming and lemmatization in the clustering of finished text documents. En Conference on Information and Knowledge Management (Washington, D.C., USA,), 625 – 633.

- Kummamuru K., Lotlikar R., Roy S. 2004. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. WWW Conferences Archive. 658-665

- Maldonado Naude F. 2002. Hermes: Servidor y biblioteca de modelos de recuperación de información. Tesis de licenciatura. Departamento de Ingeniería en Sistemas Computacionales. Escuela de Ingeniería. Universidad de las Américas, Puebla. Diciembre

- Nielsen J. 2000. *Designing web usability: The practice of simplicity*. New riders publishing. Indianapolis, United States.

- Oren Z., Oren E. 1999. Grouper: A Dynamic Clustering Interface to Web Search Results. *The Eighth Internacional World Wide Web Conference* (WWW8, Mayo, Toronto, Canada).

- Porter, M. F. 1980. An algorithm for suffix stripping. Program automated library and information systems, 14(3), 130-137.

- Salton G. Fox E.A. Wu H. 1983. Extended boolean information retrieval. *Communications of ACM*. November. 1022-1036

- Strzalkowsky, T. 1994. Robust text processing in automated information retrieval. Reading in information retrieval, K. Sparck y P. Willet Ed. Morgan Kaufmann publishers, San Francisco, California, 317-322.

- Zazo Rodríguez A. F, Figueroa Paniagua C. G, Alonso Berrocal J. L. 2002. Recuperación de información utilizando el modelo vectorial. Departamento de Informática y Automática Universidad de Salamanca. Mayo. Reporte técnico.

Enlaces de referencia

[URL01] The Dublin Core Metadata Initiative (DCMI). Disponible en: http://dublincore.org/, consultado el 05/03/2005.

[URL02] Vívisimo. Vivísimo//Vivisimo clustering – automatic categorization and meta-search software. Disponible en: http://vivisimo.com/, consultado el 06/08/2005.

[URL03] Hinrich Schütze. Single-Link, Complete-Link & Average-Link Clustering. Disponible en http://www-csli.stanford.edu/~schuetze/, consultado el 06/12/2004

Apéndice A. Conjunto de palabras vacías

Las palabras vacías consideradas en CREADOC para el Inglés y Español se muestran en las listas siguientes.

Palabras vacías para Inglés

a	come	hereupon	namely	she'll	unless
about	contains	hers	neither	she's	unlike
above	could	herself	never	should	unlikely
abst	couldn't	he's	neverthless	shouldn't	until
accordance	cry	hi	new	side	up
according	de	him	next	since	upon
across	describe	himself	nine	sincere	us
act	detail	his	ninety	six	used
actually	date	home	no	sixty	using
adj	did	how	nobody	so	very
after	didn't	however	none	some	via
afterwards	do	hundred	noone	somehow	vol
again	does	i	nonetheless	someone	was
against	doesn't	id	noone	something	wasn't
all	don't	ie	nor	sometime	way
almost	done	if	not	sometimes	we
alone	down	ill	nothing	somewhere	web
along	due	i'm	now	still	we'd
already	during	in	nowhere	stop	well
also	each	inc	o	such	were
althought	eg	include	of	take	weren't
always	eight	includes	off	taking	we've
am	eighty	indeed	often	ten	what
among	either	information	on	than	whatever
amongst	else	instead	once	that	what'll
amoungst	elsewhere	interest	one	that'll	what's
an	empty	internet	ones	that's	what've
and	end	into	only	that've	when
announce	ending	is	onto	the	whence
another	enough	isn't	or	their	whenever
any	etc	it	ord	them	where
anyhow	even	its	other	themselves	whereafter
anyone	ever	itself	others	then	whereas
anything	every	i've	otherwise	thence	whereby

anyway
anywhere
are
aren't
around
as
at
auth
available
back
be
became
because
become
becomes
becoming
been
before
beforehand
begin
beginning
behind
being
below
beside
besides
between
beyond
bill
billion
both
bottom
but
by
call
came
can
cannot
can't
caption
co
everyone
everything
everywhere
except
few
fifteen
fifty
fill
find
fire
first
five
for
former
formerly
forty
found
four
from
further
furthermore
get
give
go
got
had
has
hasn't
have
haven't
having
he
he'd
he'll
hence
her
here
hereafter
hereby
herein
heres
just
keys
last
later
latter
latterly
least
less
let
let's
like
likely
line
links
ltd
made
make
makes
many
may
maybe
me
meantime
meanwhile
might
mill
million
mine
miss
more
moreover
most
mostly
move
mr
mrs
much
must
my
myself
name
our
ours
ourselves
out
over
overall
own
page
pages
paper
part
per
perhaps
please
pp
proud
publish
put
rather
re
recent
recently
related
research
said
same
search
sec
section
see
seem
seemed
seeming
seems
seen
server
seven
seventy
several
she
she'd
there
thereafter
thereby
thered
therefore
therein
there'll
there're
there's
thereupon
these
they
they'd
they'll
they're
they've
thick
thin
third
thirty
this
those
though
thousand
three
through
throughout
thru
thus
to
together
too
top
toward
towards
trillion
twelve
twenty
two
un
under
wherein
where's
whereupon
wherever
whether
which
while
whither
who
who'd
whoever
whole
who'll
whom
whomever
who's
whose
why
will
with
within
without
won't
would
wouldn't
www
yes
yet
you
you'd
you'll
your
you're
yours
yourself
yourselves
you've

Palabras vacías para Español

a	cosas	ellos	más	por	Sra
ac	creo	es	mayor	porque	Sres
actualmente	cual	esa	me	porqué	Sta
adelante	cuales	esas	mediante	posible	Srita
ademas	cualquier	ese	mejor	próximo	su
además	cualquiera	esos	menciono	próximos	sus
afirmo	cualquieras	esta	menos	primer	suya
agrego	cuan	esta	mi	primera	suyas
ahí	cuando	estan	mia	primero	suyo
ahora	cuanta	estar	mientras	primeros	suyos
ajena	cuantas	estara	mio	principalmente	tal
ajenas	cuanto	estas	misma	propia	tales
ajeno	cuantos	este	mismas	propias	también
ajenos	cuatro	estos	mismo	propio	tampoco
al	cuenta	estoy	mismos	propios	tan
algo	da	estuvo	mucha	pudo	tanta
algún	dado	ex	muchas	puebla	tantas
alguna	dan	existe	muchisima	pueda	tanto
algunas	dar	existen	muchisimas	puede	tantos
alguno	debido	explico	muchisimo	pueden	te
algunos	de	expreso	muchisimos	pues	tener
allá	debe	fin	mucho	que	tenía
alla	deben	fue	muchos	qué	tendra
allí	decir	fuera	muy	quedo	tendrán
alrededor	dejar	fueron	nada	querer	tenemos
ambos	dejo	gran	nadie	queremos	tener
americas	del	grandes	ni	quien	tenga
ante	demas	ha	ningun	quien	tengo
anterior	demasiada	había	ningún	quienes	tenido
antes	demasiadas	habían	ninguna	quienesquiera	tercera
apenas	demasiado	haber	ningunas	quienquiera	ti
aproximadamente	demasiados	habra	ninguno	quiere	tiene
aquel	demás	hace	ningunos	realizó	tienen
aquella	dentro	hacen	no	realizado	toda
aquellas	desde	hacer	nos	realizar	todas
aquello	despues	hacerlo	nuestra	respecto	todavía
aquellos	dice	hacia	nosotras	ser	todo
aqui	dicen	haciendo	nosotros	si	todos
aquí	dicho	han	nuestro	sí	tomar
así	dieron	hasta	nuestros	siempre	total

aseguro	diferente	hay	nueva	sin	tras
aun	diferentes	haya	nuevas	solo	trata
aunque	dijeron	he	nuevo	sólo	traves
ayer	dijo	hecho	nuevos	se	tres
bajo	dio	hemos	nunca	sea	tuvo
bien	donde	hicieron	o	sean	tuya
buen	dos	hizo	ocho	según	tuyo
buena	durante	hoy	os	segunda	tu
buenas	e	hubo	otra	segundo	última
bueno	ejemplo	igual	otras	seis	ultimas
buenos	el	incluso	otro	ser	último
como	el	indico	otros	será	ultimos
cada	ella	informo	palabras	serán	un
casi	ellas	jamás	para	sería	una
cerca	ello	junto	parecer	si	unas
cierta	ellos	juntos	parte	sí	universidad
ciertas	embargo	la	partir	sido	unos
cierto	en	lado	pasada	siempre	usted
ciertos	encuentra	las	pasado	siendo	ustedes
cinco	entonces	le	pero	siete	va
clave	entre	les	pesar	sigue	vamos
comento	era	llego	poca	siguiente	van
como	eran	lleva	pocas	sin	varias
cómo	esta	llevar	poco	sino	varios
con	está	lo	pocos	sobre	veces
conmigo	estas	los	podemos	sola	ver
conocer	estás	los	podra	solamente	vez
considero	este	luego	podran	solas	vosotras
considera	estos	lugar	podrán	solo	vosotros
consigo	el	manera	podría	solos	y
contigo	ella	manifesto	podrían	son	ya
contra	ellas	mas	poner	Sr	yo

Apéndice B. Archivo de configuración

Para dar mayor flexibilidad a la interfaz del sistema, se utilizan Servlets, los cuales permiten controlar la interfaz por medio de un archivo de configuración. En el CREADOC, dicho archivo es *LocalStringC.properties.*

A continuación, por medio de un ejemplo se explicará cómo cambiar la interfaz del sistema. La Figura B.1 muestra la pantalla para clasificar documentos, a la cual se modificarán las etiquetas de *soporte Grupal mínimo y Soporte Global mínimo* por las etiquetas *CS* y *GS* respectivamente, el título del botón, el tamaño de las fuentes de los comentarios y el color de fondo.

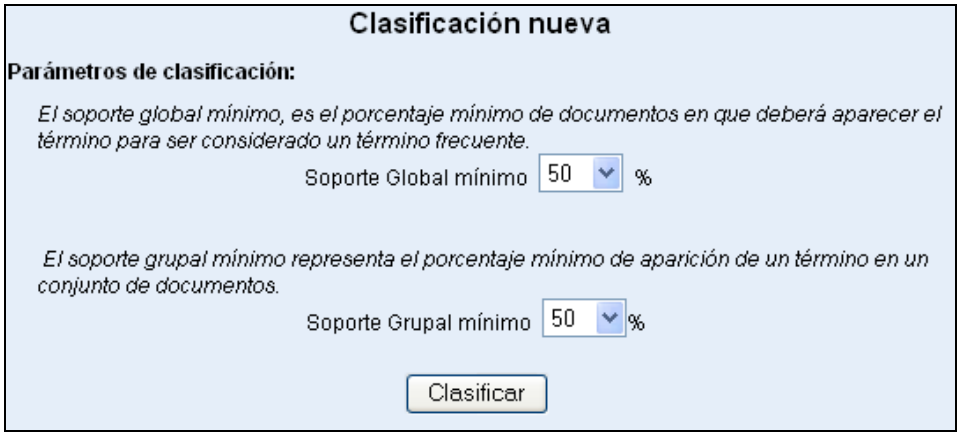


Figura B.1. Pantalla de *clasificación nueva*.

Lo primero es ubicar el archivo de configuración, el cual se encuentra en un directorio relativo a la instalación de Tomcat, por omisión es C:\Tomcat\webapps\MyServlets\WEB-INF\classes. Una vez abierto, se busca la línea que contiene: *#CLASIFICACION NUEVA,* donde el # indica que es comentario*.* Las líneas posteriores a está, hasta el siguiente #, corresponden a los valores modificables de la presente pantalla. Los cuales son los siguientes:

- CD.title=*Clasificación nueva*
- CD.titleSize=*3*
- CD.background=*rgb(229, 238, 249)*
- CD.font=*arial*
- CD.fontSize=*12px*
- CD.generalInstruction= *Parámetros de clasificación:*
- CD.askData1=*Soporte Global mínimo*
- CD.askData2=*Soporte Grupal mínimo*

- CD. button =*Clasificar*
- CD.comment1=*El soporte global mínimo, es el porcentaje mínimo de documentos en que deberá aparecer el término para ser considerado un término frecuente.*
- CD.comment2=*El soporte grupal mínimo representa el porcentaje mínimo de aparición de un término en un conjunto de documentos.*
- CD.commentFont=*italic*
- CD.commentSize=*12px*

Ahora procedemos a ubicar el nombre designado a cada variable. El color de fondo se denomina background, el título *title*, las fuentes *font*, los comentarios *comment*, los botones *button,* los tamaños S*ize* y los datos solicitados como *askData*. Todos los valores de las variables se encuentran especificados en código html, por lo que se requiere tener conocimiento del lenguaje.

Si se desean modificar los mensajes donde se solicitan los parámetros de soporte global y grupal, el mensaje del botón, el tamaño de los comentarios y el color de fondo, sustituya los valores deseados que se muestran en negritas.

- CD.title=*Clasificación nueva*
- CD.titleSize=*3*
- CD.background=***rgb(255, 255, 255)***
- CD.font=*arial*
- CD.fontSize=*12px*
- CD.generalInstruction= *Parámetros de clasificación:*
- CD.askData1=***GS***
- CD.askData2=***CS***
- CD. button=***Iniciar clasificación***
- CD.comment1=*El soporte global mínimo, es el porcentaje mínimo de documentos en que deberá aparecer el término para ser considerado un término frecuente.*
- CD.comment2=*El soporte grupal mínimo representa el porcentaje mínimo de aparición de un término en un conjunto de documentos.*
- CD.commentFont=*italic*
- CD.commentSize=***10px***

Una vez realizados y guardados los cambios se debe reiniciar el contenedor de Servlets (Tomcat). Al navegar de nuevo a la pantalla, ahora se mostrará una interfaz como la mostrada en la Figura B.2.

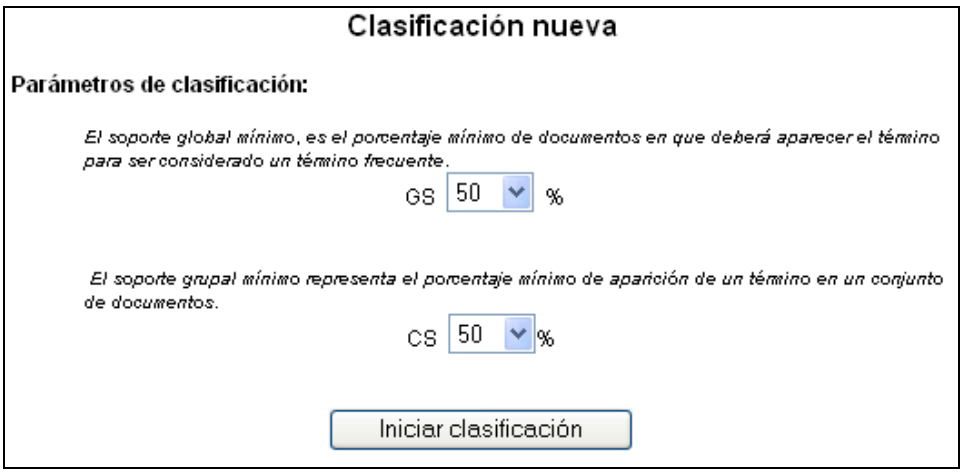


Figura B.2. Pantalla de *clasificación nueva* modificada.

Glosario

- *Applet.* Se refiere a un programa pequeño escrito en Java que se ejecuta en un navegador web

- *Application Programming Interface (API)*. Interfaz de Programación de Aplicaciones, es un conjunto de especificaciones de comunicación entre componentes software. Representa un método para conseguir abstracción en la programación

- *CREADOC.* Clasificación y Recuperación Automática de Documentos, sistema que implementa técnicas de clasificación y recuperación de documentos sin supervisión humana

- *Dublin Core Metadata Initiative (DCMI)*. Iniciativa de Metadatos Dublin Core, es una organización dedicada a la promoción y difusión de normas interoperables sobre metadatos y desarrollo de vocabularios especializados para describir recursos

- *Java Runtime Enviroment (JRE)*. Entorno de Ejecución Java donde se ejecutan las aplicaciones desarrolladas en lenguaje Java

- *Frequent Item-based Hierarchical Clustering (FIHC).* Método de agrupamiento de documentos cuyo significado se traduce como agrupamiento basado en conjunto de objetos frecuentes

- *Frequent Itemsets (FI).* Objetos frecuentes, se refiere a un conjunto de términos que ocurren en una fracción mínima de documentos.

- *Hierarchical Frequent Term-based Clustering (HFTC)*. Método de agrupamiento jerárquico basado en términos frecuentes, es un algoritmo difuso con agrupamiento jerárquico de términos comunes

- *Intellectual Property Rights (IPR)*. Derechos de Propiedad Intelectual

- *Inverse Document Frequency (IDF)*. Frecuencia Inversa de Documentos, formalmente definida como el logaritmo del conciente entre el número total de documentos y el número de documentos que contienen el término buscado

- *Metabuscador*. Buscador de buscadores. Envía la búsqueda solicitada a varios buscadores simultáneamente y responde con las direcciones localizadas en todos ellos

- *MySQL.* Manejador de bases de datos más populares desarrolladas bajo la filosofía de código abierto

- *Palabras claves.* Conjunto de términos que describen el contenido de un documento. Generalmente son los términos con mayor frecuencia de aparición dentro del documento

- *Palabras vacías.* Conjunto de términos no significativos tales como artículos, preposiciones, adjetivos, que se consideran irrelevantes al realizar búsquedas

- *Retrieval Information (RI).* En Español, recuperación de información

- *Servlet.* Programa pequeño escrito en Java que se ejecuta en un servidor web. Los Servlets son objetos que extienden su funcionalidad al utilizar el contexto de un contenedor de Servlets.

- *Soporte global mínimo (GS).* Porcentaje mínimo de documentos que deberán contener a un término para que éste se considere término frecuente

- *Soporte grupal mínimo (CS).* Porcentaje mínimo de aparición de un término en un grupo de documentos

- *Suffix Tree Clustering (STC)*. Agrupamiento de árbol de sufijos, es un método de agrupamiento que se basa en conjuntos de documentos que comparten frases comunes

- *Term Frequency (TF).* Frecuencia de un término, número de apariciones de un término en un documento

- *Tomcat (Jakarta Tomcat)*. Funciona como un contenedor de Servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los Servlets y de JavaServer Pages (JSP) de Sun Microsystems. Se considera un servidor de aplicaciones

- *Unified Modelling Language (UML).* Lenguaje unificado de modelado, permite modelar sistemas de software

- *Usuario.* Cualquier persona que interactúe con el sistema