

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECUPERACIÓN DE  
INFORMACIÓN GRÁFICA BASADO EN LA METODOLOGÍA MAS-CommonKADS

LUIS ARMANDO PORTILLA GRANADOS

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA  
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY  
MAESTRÍA EN CIENCIAS COMPUTACIONALES  
BUCARAMANGA  
2005

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECUPERACIÓN DE  
INFORMACIÓN GRÁFICA BASADO EN LA METODOLOGÍA MAS-CommonKADS

LUIS ARMANDO PORTILLA GRANADOS

Trabajo de Grado

Director  
Dr. ÁLVARO ENRIQUE ARENAS DPhil

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA  
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY  
MAESTRÍA EN CIENCIAS COMPUTACIONALES  
BUCARAMANGA  
2005

Nota de aceptación:

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Bucaramanga, 15 de julio de 2005

## CONTENIDO

	Pág
INTRODUCCIÓN	1
1. EL PROBLEMA	2
1.1 TÍTULO	2
1.2 DESCRIPCIÓN DEL PROBLEMA	2
1.3 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	2
1.4 ANTECEDENTES	3
1.5 OBJETIVOS	9
1.5.1 Objetivo general	9
1.5.2 Objetivos específicos	9
1.6 RESULTADOS ESPERADOS	9
2. MARCO TEÓRICO	12
2.1 INTRODUCCIÓN	12
2.2 AGENTES	13
2.3 INGENIERÍA DE SOFTWARE ORIENTADA A AGENTES (AOSE)	15
2.4 INGENIERÍA DEL CONOCIMIENTO	17
2.5 METODOLOGÍAS ORIENTADAS A AGENTES	17
2.5.1 Análisis y diseño OA de Burmesister	17

2.5.2	Técnica de modelado de agentes para sistemas de agentes BDI	18
2.5.3	Método basado en escenarios multiagente (MASB)	19
2.5.4	Metodología orientada a agentes para modelado de empresas	20
2.5.5	Metodología CommonKADS	20
2.5.6	Metodología CoMoMAS	22
2.6	METODOLOGÍA MAS-COMMONKADS	23
2.6.1	Modelos de MAS-CommonKADS	23
2.6.2	Fases de desarrollo de MAS-CommonKADS	26
2.7	Recuperación de Información	26
2.7.1	Modelos de recuperación	29
2.7.2	Evaluación de la recuperación	34
2.8	PROCESAMIENTO DE TEXTO	35
2.9	BASES DE DATOS DE IMÁGENES	36
2.9.1	Definición de bases de datos	36
2.9.2	Almacenamiento	38
2.9.3	Tipos de almacenamiento	39
2.9.4	Necesidades de almacenamiento	39
2.9.5	Formatos gráficos	40
2.9.6	Perfil	42
2.9.7	¿Qué es un perfil?	42

2.9.8	Aplicaciones del Profiling	43
3.	ANÁLISIS	47
3.1	INTRODUCCIÓN	47
3.2	CONCEPTUACIÓN	48
3.2.1	Enunciado del problema	48
3.2.2	Identificación de los actores	49
3.2.3	Descripción de los actores	49
3.2.4	Identificación de los casos de uso	49
3.2.5	Descripción de los casos de uso	51
3.3	MODELO DE AGENTE	54
3.3.1	Identificación de los agentes	54
3.3.2	Descripción de los agentes	55
3.4	MODELO DE TAREAS	56
3.5	MODELOS DE COORDINACIÓN, COMUNICACIÓN, ORGANIZACIÓN	62
3.5.1	Identificación de conversaciones	63
3.5.2	Descripción de las conversaciones	63
3.6	MODELO DE EXPERIENCIA	65
4.	DISEÑO	67
4.1	INTRODUCCIÓN	67
4.2	DISEÑO DE LOS AGENTES: DESCOMPOSICIÓN EN SUBSISTEMAS	67

4.2.1	Subsistema para la asignación de perfiles	67
4.2.2	Subsistema para el procesamiento de texto	69
4.2.3	Subsistema para la administración de la base de datos y RIG	74
4.3	DISEÑO DE LA INTERFAZ GRÁFICA DE USUARIO	76
4.3.1	Administrador BDatos (Editor RIG)	76
4.3.2	Recuperación de Información Gráfica (RIG)	80
4.4	DISEÑO DE LA BASE DE DATOS DEL SISTEMA	84
4.4.1	Modelo	84
4.4.2	Descripción del modelo	85
4.5	DISEÑO DE LA PLATAFORMA	89
5.	DESARROLLO	90
5.1	Introducción	90
5.2	Declaración de clases	92
5.3	Clase Edit RIG (Agente Bdatos)	93
5.3.1	Principal	93
5.3.2	Categorías	94
5.3.3	Palabras genéricas, derivadas y sinónimas	94
5.3.4	Imágenes y atributos	95
5.3.5	Reinicio de pesos	95
5.4	Clase RIG (Agente Recuperación de Información Gráfica)	96

5.4.1	Principal	96
5.4.2	Procesamiento de texto	97
5.4.3	Resultados de recuperación	97
5.4.4	Configuración de perfiles	98
5.4.5	Historial	98
5.5	DESARROLLO DE LA INTERFAZ CON EL USUARIO	99
5.6	ARCHIVOS ADJUNTOS	99
	CONCLUSIONES	100
	RECOMENDACIONES	102
	BIBLIOGRAFÍA	103



## TABLA DE FIGURAS

	Pág.
Figura 1. El esquema de pantalla de Aria	7
Figura 2. La documentación de la imagen	8
Figura 3. El conjunto de modelos de CommonKADS	21
Figura 4. Modelos metodología MAS-CommonKADS	25
Figura 5. Sistema de recuperación de información	27
Figura 6. Modelos de recuperación de información	30
Figura 7. Diagrama de flujos agente usuario	57
Figura 8. Diagrama de flujos agente Bdatos	59
Figura 9. Diagrama de flujos agente Recuperación	61
Figura 10. Caso de uso interno	63
Figura 11. Ventana editor RIG	76
Figura 12. Ventana para edición de categorías	77
Figura 13. Ventana edición de palabras genéricas, derivadas y sinónimas	78
Figura 14. Ventana para edición de información de imágenes	79
Figura 15. Ventana demostrativa de pesos	79
Figura 16. Ventana Recuperación de Información Gráfica – RIG	80
Figura 17. Ventana para procesamiento de texto	81
Figura 18. Ventana de presentación de imágenes exitosas	82
Figura 19. Ventana para la configuración de perfiles	83
Figura 20. Ventana de historial de usuario	84
Figura 21. Modelo entidad relación	85

## **RESUMEN**

El trabajo plantea el desarrollo de un prototipo para la Recuperación de Información Gráfica (RIG) a partir de su diseño aplicando la metodología orientada a agentes MAS-CommonKADS y utilizándose el modelo vectorial para la recuperación de información. Este prototipo utiliza perfiles tanto de usuario como de trabajo, además de las herramientas para el procesamiento de texto que determina palabras relevantes a partir de un texto base, generando resultados exitosos y logrando un "aprendizaje" a partir de la experiencia de su uso y del concepto gráfico de los usuarios. Para lograrlo se soporta en teorías que describen las principales metodologías de programación orientadas a agentes y sus principales consideraciones para su diseño y desarrollo.

## **ABSTRACT**

This thesis deals with the development of a Prototype for Graphic Information Retrieval (GIR). It applies the agent-oriented methodology MAS-CommonKADS and the use of a vectorial model for retrieving information. This prototype employs profiles of either a user or a work, moreover it uses the tools for the text processing that determines relevant words from a base text, producing successful results and achieving a "Learning" out of the experiences of its use and the graphic concept of users. To achieve this, it based on theories which describe the main programming methods towards agents and its main ideas to design and develop them.

## **INTRODUCCIÓN**

La evolución continua y progresiva de la tecnología hace que los sistemas sean cada vez más precisos, eficientes y de alguna forma inteligentes, para que los usuarios finales no se vean saturados debido al crecimiento diario que la información experimenta.

Las nuevas tecnologías y los nuevos sistemas de información demandan la utilización de agentes de software autónomos dentro del campo de la Inteligencia Artificial para hacer posible aplicativos tan sofisticados que pueden lograr ser entidades independientes y completas en sí mismas, como tan sencillos que se pueden convertir en una herramienta útil a todo tipo de usuario que requiera información sin conocimientos previos de sistemas.

Lo ideal es contar con aplicativos que permitan la selección autónoma y exitosa de información, para este caso información gráfica, específica para diferentes ámbitos: editoriales, investigación, periodismo, publicidad, medicina, materiales educativos, etc., desde el punto de vista técnico como documental permitiendo aprovechar, con la automatización de servicios, los más adecuados recursos disponibles en los bancos de información gráfica.

La tecnología de agentes está recibiendo una gran atención en los últimos años, sin embargo, a pesar del rápido desarrollo de teorías, arquitecturas y lenguajes de agentes, se ha realizado muy poco trabajo en desarrollar aplicaciones empleando ésta tecnología.

La propuesta entonces aborda la creación de un aplicativo de recuperación de información gráfica que permita hacer uso racional y eficiente de las nuevas tecnologías en el campo editorial.

El documento se compone de cinco capítulos: el primero aborda el problema, los objetivos y los resultados esperados; el capítulo dos examina el estado del arte alrededor de la recuperación de información y temas asociados; el tercero corresponde al análisis aplicando la metodología MAS-CommonKADS y presenta la conceptualización y los modelos de agente, tareas, coordinación, comunicación, organización y experiencia; el cuatro capítulo estructura las decisiones de diseño conducentes a definir los requisitos de implementación; y el quinto capítulo se dedica al desarrollo del prototipo.

## **1. EL PROBLEMA**

### **1.1 TÍTULO**

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECUPERACIÓN DE INFORMACIÓN GRÁFICA BASADO EN LA METODOLOGÍA MAS-CommonKADS

### **1.2 DESCRIPCIÓN DEL PROBLEMA**

Con la consecución del presente proyecto se pretende el diseño y desarrollo de una aplicación prototipo que permita facilitar el trabajo del usuario recuperando por demanda las imágenes que este requiere y/o necesite en la preparación de materiales sobre cualquier editor con sólo activar el aplicativo, tomando como referencia de consulta una selección de texto o ingresando al aplicativo directamente atributos de consulta, y que adicionalmente para cualquiera de los casos se permitirá determinar perfiles de usuario y de trabajo para delimitar el grupo de posibles imágenes exitosas, permitiendo además operaciones básicas sobre los documentos como introducir nuevos documentos, modificarlos o eliminarlos, y operaciones sobre la documentación indexada para realimentar y modificar las características iniciales.

El aplicativo aporta además de lo anterior ahorro de sistema en el sentido de que sólo se activará cuando el usuario lo solicite, deja a total disposición los recursos de máquina para su trabajo, igualmente que descongestiona visualmente el área de trabajo, y está pensado para su aplicación en diferentes áreas donde se hará cada vez más experto en la medida que se utilice.

### **1.3 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN**

Nuestra sociedad actual está protagonizada por la imagen, en informática los entornos son cada vez más gráficos, en ilustraciones editoriales, libros electrónicos entre muchos otros campos, el empleo de imágenes como recurso informativo es cada vez mayor en multitud de organizaciones y servicios de información de distintos ámbitos: editoriales, investigación, periodismo, publicidad, medicina, educación, etc. Ante esta situación, los bancos de imágenes de los medios de comunicación, desarrolladores para Internet, empresas editoriales y fototecas ven aumentadas sus colecciones de manera exponencial.

Los avances en las técnicas y procedimientos de digitalización y almacenamiento, unidos al aumento y mejora de aplicaciones de gestión documental multimedia,

conlleven un abandono progresivo de la bases de datos referenciales, permitiendo replantear la organización y recuperación en los servicios de documentación para integrar en el mismo sistema tanto el documento primario en forma de imágenes (fotografías, ilustraciones, documentos textuales, etc.) como el documento secundario (ficha de descripción), consultables de manera rápida, sencilla y contextualizada ya que permite, en la recuperación, una visualización directa del objeto de información, del tal manera que el propio usuario final puede evaluar la relevancia de los resultados obtenidos.

Generalmente estos bancos de imágenes pueden llegar a saturarse debido al crecimiento diario que experimentan, creando desventajas para el usuario final como excesiva inversión de tiempo, selecciones inadecuadas, las búsquedas se limitan a unos pocos referentes sin interacción o retroalimentación.

Como alternativa de programación los agentes nacieron de la investigación en Inteligencia Artificial (IA) y más concretamente de la Inteligencia Artificial Distribuida (DIA). Al principio, la IA hablaba de los agentes como programas especiales cuya naturaleza y construcción no se llegaba a detallar. Recientemente se ha escrito mucho acerca de este tema explicando qué se puede entender por un agente y qué no, en general atendiendo a características que debe tener un agente: autonomía, reactividad, iniciativa, habilidad social, etc.

El crecimiento observado en el uso de agentes de software para el diseño de aplicaciones así como el desarrollo presentado por la ingeniería del software orientada a agentes posibilita la aplicación de metodologías como MAS-CommonKADS en el diseño de un sistema que optimice los procesos de documentación y recuperación de información gráfica almacenada en bases de datos, permitiendo su adaptabilidad a las diferentes preferencias del usuario, así como la valoración y utilización del contexto de trabajo del usuario y sus requerimientos para el enriquecimiento de la caracterización en los bancos de imágenes.

#### **1.4 ANTECEDENTES**

La Recuperación de Información es uno de las aplicaciones más frecuentes de los sistemas de agentes, tiene uno de sus orígenes en las bibliotecas y centros de documentación en los que se requerían búsquedas bibliográficas de libros y artículos de revista. El objetivo principal de cualquier centro de documentación es satisfacer las necesidades reales y potenciales de información de todos los usuarios, proporcionándoles la información veraz, pertinente, justo a tiempo y al menor costo. El objetivo de la recuperación de información es, dada una

necesidad de información y un conjunto de documentos, ordenar los documentos de mayor a menor relevancia para esa necesidad y presentarlos al usuario<sup>1</sup>.

Con el incremento del número de documentos en formato electrónico, se hace necesario contar con herramientas informáticas adecuadas para la recuperación de documentos. Las técnicas manuales han demostrado ser ineficaces, pues básicamente consisten en la elaboración manual de una descripción del contenido temático de cada uno de los documentos, siendo éste un trabajo costoso en tiempo, que además tiene abundante inconsistencias<sup>2</sup>. La investigación en recuperación de información busca diseñar sistemas que acepten consultas en lenguaje natural y proporcionen documentos adecuados a tales consultas, ordenados según algún criterio del sistema, de acuerdo a las características de los documentos y a las necesidades informativas expresadas por el usuario en su consulta<sup>3</sup>. Por simplicidad, se utilizará el término documento para denotar a unidades genéricas de información de diversa índole, incluyendo artículos o libros impresos o microfilmados, piezas de audio o video analógicos, así como archivos digitales de texto, audio, video o gráficas que es el objeto del presente proyecto.

En una gran cantidad de trabajos se han realizado desarrollos basados en diferentes teorías y paradigmas dentro de las Ciencias Computacionales. Los sistemas expertos, con su capacidad para combinar información y reglas de actuación, han sido vistos como una de las posibles soluciones al tratamiento y recuperación de información, no sólo documental. La década de los 80's fue prolija en investigación y publicaciones sobre experimentos de este orden, interés que continua en la presente década. En resumen, un sistema experto es una aplicación capaz de realizar las tareas propias de un experto humano en un área restringida. Se compone de una base de datos, de una base de reglas y de un motor de inferencia<sup>4</sup>. La base de datos almacena el conjunto de datos o documentos sobre los que se desea ejecutar una serie de acciones. La base de reglas contiene un compendio de reglas lógicas que el sistema debe utilizar para desarrollar razonamientos, así como las normas que permiten combinar las reglas, por último, el motor de inferencia es el encargado de ejecutar las órdenes del usuario, utilizando como criterios las reglas, y como material de partida el

---

<sup>1</sup> ZAZO, Ángel, C. Figuerola, J. L. Alonso, R. Gómez. Recuperación de Información utilizando el modelo vectorial. Participación en el taller CLEF-2001. Technical Report DPTOIA-IT-2002-006. mayo, 2002.

<sup>2</sup> HOOPER, R. S. Indexer consistency tests-origin, measurements, results and utilization. Bethesda, MD, 1965.

<sup>3</sup> BELKIN, N.J.; CROFT, W.B. Retrieval techniques. Annual Review of Information Science and Technology, 22, p. 109-145, 1987

<sup>4</sup> FROST, R. Bases de Datos y Sistemas Expertos. Ingeniería del Conocimiento. Madris Díaz de Santos, 1989.

contenido de la base de datos, hasta alcanzar una conclusión simulando el razonamiento que seguiría el experto humano.

El desarrollo posterior de estos sistemas ha traído un conjunto de nuevas aplicaciones a las que se identifica como Sistemas Basados en el Conocimiento (SBC), que incorporan técnicas más sofisticadas como la lógica difusa, razonamiento basado en modelos, etc., y lo que puede ser más interesante para el especialista en información, medios de recuperación de información deductiva como lo expone FROST<sup>5</sup>.

Con el dramático crecimiento de la Internet, las redes han cambiado el papel de las computadoras, dejando de ser simples procesadores de datos aislados y convirtiéndose en poderosos dispositivos de comunicación. Los usuarios ahora pueden acceder a una vasta y amplia cantidad de información. Dentro de este contexto, surge la necesidad de proveer al usuario de medios para manejar dicha información. Generalmente el usuario se encuentra con la necesidad de navegar entre el gran volumen de datos para buscar y localizar la información que necesita y que le es de utilidad. En este escenario nacen las bibliotecas digitales que además de proveer servicios similares a las bibliotecas convencionales, tales como catalogación y búsqueda, también brindan una gran variedad de servicios adicionales, tales como proveer información de una manera organizada y facilitar al usuario el manejo de esta información dinámica y compleja. Dichas bibliotecas se han visto enormemente beneficiadas con el uso de agentes de usuario, los cuales son entidades de software autónomas o semiautónomas que realizan misiones bien definidas que son delegadas (o solicitadas) expresamente por el usuario en su beneficio y de acuerdo a sus necesidades y objetivos.

Uno de los desafíos futuros es la búsqueda de objetos multimediales (video, imágenes, música, etc) por contenido, y no por palabras en la cercanía de la imagen como existe ahora. El análisis automatizado de la imagen debería, por ejemplo, identificar personas, lugares y cosas en una fotografía, y así documentar la imagen. Aunque investigadores han logrado progresos considerables en esta área, aun se esta lejos de confiar en este tipo de acercamiento ya que muchos rasgos sobresalientes solo están en la mente del usuario. Indexar las imágenes requeridas es una manera de comunicar estas características a la máquina. El crecimiento de las bibliotecas digitales, los sistemas de información gráfica y fototecas, ha impulsado a los investigadores a enfocar sus esfuerzos en este sentido. Un ejemplo de ello es el proyecto desarrollado en conjunto, por el Instituto Tecnológico de Massachussets y Kodak Company: ARIA<sup>6</sup>.

---

<sup>5</sup> Ibid.

<sup>6</sup> LIEBERMAN, Henry. Aria: An Agent for Annotating and Retrieving Images. MIT Media Lab.

Aria es un agente de interfaz, que facilita al usuario la documentación y recuperación de imágenes de su propia base de datos. Aunque el proceso no está totalmente automatizado, el uso de Aria conduce a llevar bibliotecas de imágenes bien documentadas y maximiza la oportunidad de considerar mayor cantidad de imágenes.

Los investigadores de Kodak, observaron que el proceso de tomar y usar las fotografías en formato digital, requería más esfuerzo del que debería. Organizar y después recuperar las imágenes guardadas es tan tedioso que las personas evitan hacerlo y raramente vuelven a ver muchas de sus fotografías. Una opción es el uso de etiquetas de texto para documentar las imágenes y guardarlas en una base de datos relacional, para posteriormente recuperarla haciendo consultas por palabras, pero resulta improbable que los usuarios inviertan sustancialmente sus esfuerzos para clasificar y categorizar las imágenes esperando facilitar la recuperación futura. Además, la recuperación requiere el uso de algún tipo de motor de búsqueda o aplicación similar, lo que es poco probable, aun solo en lo que se refiere a iniciar o terminar dicho programa para introducir palabras clave. El uso de agentes de software para reemplazar estas labores podría reducir de alguna forma el aburrimiento.

Los investigadores diseñaron un agente de interface de usuario para facilitar la documentación textual y la recuperación. El papel del agente no es tanto la recuperación y documentación automática, sino detectar oportunidades para realizar estas funciones y alertar al usuario en dichas circunstancias. El agente puede hacerlo tan fácil como sea posible para que el usuario pueda completar las operaciones cuando sea apropiado.

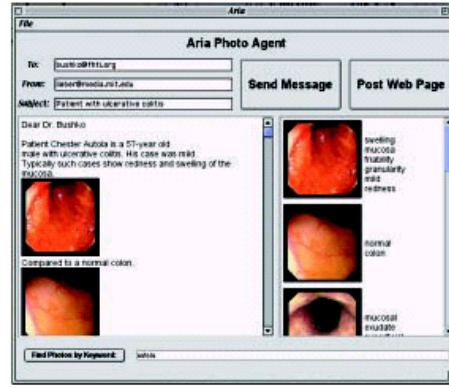
Aria integra documentación de la imagen, recuperación, y uso en una sola aplicación, eliminando de esta manera la molestia que impone el uso de las aplicaciones separadas. La integración de estas utilidades proporciona un contexto apropiado ya que relaciona el contenido escrito con las imágenes disponibles.

La aplicación inicial consiste de un editor de texto estándar de Java Swing acoplado a un panel que contiene una imagen personalizada por una aplicación de recuperación y documentación.

Al fondo del editor del texto, Aria despliega una imagen obtenida por la recuperación tomada del texto que rodea el cursor del editor. A la derecha del editor de texto una columna de imágenes recuperadas que Aria actualiza dinámicamente. A la derecha de cada imagen, la aplicación despliega una lista de palabras claves de la documentación que pertenecen a cada imagen.



Figura 1. El esquema de pantalla de Aria.



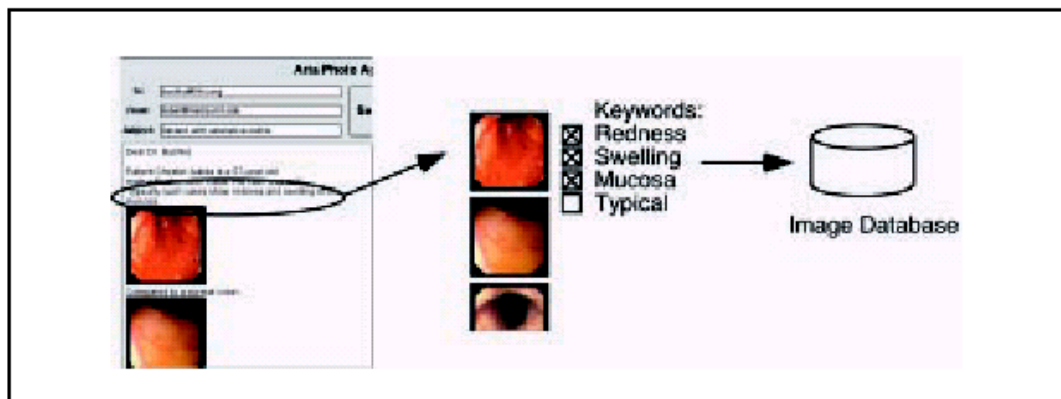
Aria se ejecuta continuamente y observa lo que el usuario está tecleando. Analiza la entrada del agente para extraer las palabras claves que rodean el cursor de texto. La aplicación acostumbra a utilizar las heurísticas comunes de extracción de información similares a aquellas usadas por los motores de búsqueda de web para realizar el análisis del texto, aunque se está experimentando con otros métodos. Además, despliega continuamente las palabras claves junto al área del cursor, en la caja debajo del editor de texto. La aplicación usa las palabras claves extraídas para preguntar a la base de datos de imágenes y despliega una lista de cuadros alineada en el orden de relevancia en la columna de la derecha. Si la imagen deseada no aparece inmediatamente, el usuario puede buscar a través de la lista hasta que una imagen conveniente aparezca o llama un cuadro de diálogo para cargar otras imágenes. Aun cuando la búsqueda puede requerir alguna de estas alternativas, todavía economiza interacciones del usuario, comparado con una aplicación convencional.

Aria también facilita el proceso de documentación de la base de imágenes. Para ello observa las palabras claves circundantes al cursor de texto, y si una imagen recuperada es seleccionada, añade a la documentación de esta imagen las palabras claves que no estaban asociadas a ella, enriqueciendo la documentación. Si no hay una imagen asociada a las palabras claves y el usuario selecciona alguna de la base de imágenes, automáticamente las palabras claves son añadidas a la documentación de la imagen elegida.

El texto que escribe el usuario no es específicamente para documentar alguna imagen, si no por ejemplo, para redactar un mensaje a otra persona. Es por ello que las suposiciones de Aria no son necesariamente correctas, por lo cual el usuario tiene siempre la opción de seleccionar o eliminar palabras que no documenten adecuadamente una imagen. Si el usuario hace clic sobre una imagen, el sistema despliega un editor que permite seleccionar el juego de

palabras claves conveniente; si el editor excluye una palabra adecuada, el usuario puede arrastrar con el mouse dicha palabra desde el editor de texto y así agregarla a la documentación. Esto suele ser más útil que la documentación tradicional de la imagen.

Figura 2. La documentación de la imagen.



Finalmente, a nivel de productos<sup>7</sup> para la gestión de sistemas de información gráfica, los aplicativos que se encuentran en el mercado son extensiones de sistemas de bases de datos documentales tradicionales o sistemas de Gestión Electrónica de Documentos (GED) con todas sus prestaciones (búsquedas booleanas, aproximación, gestión de tesauros, etc.), que integran visualizadores que determinarán la forma de presentación y selección de los resultados de una búsqueda y ofrecerán una serie de herramientas para su manipulación: zoom, rotación, etc. Éste es el caso de GES'IMAGES comercializado por la casa Chemdata Ibérica, que integra el núcleo documental Taurus.

A nivel de sistema de gestión de información de aplicación general, la casa Chemdata Ibérica hoy día conocida como EVER documéntica, gestión de la información y del conocimiento, que es filial del grupo EVER, líder francés en sistemas de Gestión Documental y ECM (Enterprise Content Management), con productos para la empresa como sistemas de información documental, gestores de biblioteca y gestores de archivos, dedicada a la gestión y tratamiento de la Información, Documentación y Conocimiento en sus diferentes vertientes.

De forma similar se cuenta con el sistema de gestión documental del Grupo GESFOR de España, Tecnología de Información, que está presente en el mercado con sistema de gestión documental integral que gestiona todo tipo de objetos,

<sup>7</sup> GARCÍA CABALLERO, Ricardo. Servicio de Documentación Gráfica. Everes S. A. Nuevas Tecnologías al Servicio de la Información Gráfica. FESABID

como gráficos, artículos de prensa, fotografías, documentos, entre otros. Esta solución permite la creación y desarrollo de bibliotecas, fototecas, recortes de prensa, salas virtuales de arte, etc., basándose en herramientas como gestión genérica de documentos, que permite una organización jerárquica de los objetos, y su búsqueda y localización mediante catálogos de base de datos, herramientas de procesos y auditorias, entre otras, igualmente de aplicación empresarial.

## **1.5 OBJETIVOS**

### **1.5.1 Objetivo general**

Diseño e implementación de un prototipo para la recuperación de información gráfica utilizando la metodología MAS-CommonKADS.

### **1.5.2 Objetivos específicos**

- Analizar las últimas técnicas empleadas en la recuperación de información y la pertinencia de su aplicación en el prototipo a desarrollar utilizando los paradigmas de agentes.
- Estudiar las técnicas recientes de documentación en bases de datos de información gráfica para adaptarlas en el diseño y desarrollo de un prototipo para recuperación de información gráfica.
- Aplicar la metodología MAS-CommonKADS para el diseño de un prototipo para la recuperación de información gráfica.
- Desarrollar un prototipo del sistema de recuperación de información gráfica.

## **1.6 RESULTADOS ESPERADOS**

- Estado del arte sobre metodologías y programación orientada a agentes, procesamiento de texto y recuperación del información.
- Aplicación de la metodología MAS-CommonKADS para el diseño de un sistema de recuperación de información gráfica.
- Implementación de un prototipo del sistema de recuperación de información gráfica, utilizando el lenguaje de programación JAVA. Un aplicativo capaz de:
  - Personalizar el aplicativo: Crear, modificar o eliminar perfiles de usuario y de área de trabajo.

- Procesar texto.
  - Gestionar una base de datos de información gráfica.
  - “Aprender” de la interacción con los usuarios.
  - Actualizar información automáticamente o por retroalimentación.
  - Presentar resultados de la recuperación.
- Como novedades se pueden considerar los siguientes aspectos:
    - La aplicación, de manera rigurosa, de una metodología orientada a agentes como MAS-CommonKADS en un nuevo dominio como es el caso de la recuperación de información gráfica.
    - La personalización del aplicativo que no solamente define sus usuarios sino que permite asociarle diferentes áreas de trabajo y categorías de imágenes según los requerimientos del material.
    - Conocer paso a paso los eventos y resultados de los procesos ejecutados.
    - Que el sistema además del “aprendizaje” producto de la aplicación del modelo vectorial, se enriquece del concepto gráfico que posean sus usuarios.
    - Desarrollo utilizando programación orientada a objetos con desarrollo en herramientas de Java.

Como una aporte interesante a esta sección cabe agregar una comparación entre el prototipo de Recuperación de Información Gráfica desarrollado y el prototipo Aria, ya que Aria fue el sistema base para el diseño y desarrollo del proyecto por ajustarse a las necesidades y objetivos planteados, por su particularidad en representar un agente que integre recuperación y documentación.

Como similitudes en términos generales se puede anotar que se trata de dos prototipos basados en agentes; que tienen la particularidad de que sus resultados de recuperación son imágenes, a diferencias del común de los sistemas de recuperación que se enfocan en la recuperación de referencias documentales como es el caso de buscadores temáticos, las bibliotecas digitales y gestión de archivo; que los prototipos utilizan bases de datos de imágenes que incluyen documentación con atributos de cada imagen; que los dos se basan en el procesamiento de texto que permiten a partir de un grupo de palabras determinar palabras clave para realizar comparaciones en espera de coincidencias en la documentación de las imágenes y así predeterminedar las imágenes relacionadas para su selección; similar también las políticas para la presentación de los resultados por orden de relevancia; y en la metodología para alimentar su documentación.

Igualmente es relevante determinar las diferencias entre el Sistema de Recuperación de Información Gráfica y el Sistema Aria en múltiples aspectos de los que cabe destacar de forma general los siguientes:

- La documentación de las imágenes esta controlada permitiendo sólo palabras genéricas eliminando grupos de palabras semánticamente diferentes pero con el mismo significado (tesauros y lematización).
- La recuperación se hace por demanda, se ejecuta sólo cuando el usuario requiera una imagen, evitando gasto innecesario de sistema, además que funcionalmente no todo lo que se escribe requiere de ser ilustrado.
- Que el prototipo esta diseñado para cualquier tipo de imagen y permite su utilización en varias áreas, que para el caso Aria esta pensado a nivel de fotografía y más específicamente para el área de medicina.
- Que permite administrar la base de datos.
- Que el sistema permite guardar un historial y el comportamiento de todos sus componentes como pesos, perfiles, etc.
- El sistema se puede ejecutar desde cualquier aplicativo y no estaría circunscrita a un editor específico ya que la mayoría de los materiales a graficar estarán sujetos a un software determinado de acuerdo a los requerimientos técnicos y a las costumbres y habilidades del usuario.
- Y por último, como uno de los elementos más importantes, la personalización que dota al prototipo de posibilidades de diferentes usuarios, ya que según el criterio o concepto gráfico de cada uno puede cambiar su aplicabilidad y por consecuencia su "aprendizaje"; así como la personalización por áreas de trabajo que asocia una serie de categorías de imágenes a cada perfil delimitando la búsqueda. Los perfiles tanto de usuario como de trabajo son flexibles y permiten crecimiento y adaptación que ajusta el aplicativo a todos los requerimientos de trabajo.

Por otro lado existen diversos trabajos sobre recuperación de información gráfica, sistemas de recuperación de imágenes, gestores de información y bases de datos de imágenes, orientados a metodologías, recursos, estructuras de bases de datos; en variadas aplicaciones como la televisión, el periodismo, la medicina, la industria, la empresa y muchas más que son irrelevantes para el objeto del presente trabajo por su naturaleza y campo de acción.

## 2. MARCO TEÓRICO

### 2.1 INTRODUCCIÓN

Este capítulo no tiene como fin la creación de nuevas definiciones sino que se apoyará en fuentes de información para apuntar las propiedades que serán más importantes a la hora de abordar la complejidad del sistema y su desarrollo.

Se parte de la teoría de agentes, como un nuevo enfoque a los problemas distribuidos, dando lugar a metodologías capaces de enfrentar dichos problemas de forma autónoma, gracias a sus elementos constitutivos y a las funciones que cada uno aplica en búsqueda de ese objetivo común. De estas metodologías se aplicará MAS-CommonKADS para la creación del prototipo.

Paralelo al componente metodológico se trabaja para ubicar la teoría de agentes al contexto del sistema, haciéndose necesaria la incorporación de contenidos acerca de una de sus aplicaciones de forma general como lo es la recuperación de información, ofreciendo una visión general del proceso, desde la indexación de la colección de documentos, hasta finalmente la evaluación de la recuperación en sistemas experimentales; se introducirá el concepto de modelo de recuperación y se expondrá los más utilizados tomando el modelo vectorial como base para el presente trabajo y que se construirá y detallará en los siguientes capítulos de análisis y diseño, adaptando dichos contenidos al uso de información gráfica.

Este componente conceptual se complementa con las herramientas para el procesamiento de texto, con las bases datos de información gráfica, además de un componente sobre perfiles, como fundamentos para el diseño y desarrollo de un prototipo de recuperación de información gráfica.

### 2.2 AGENTES

Se puede encontrar propuestas en la literatura de un gran número de definiciones del concepto de agente, sin que ninguna de ellas haya sido plenamente aceptada por la comunidad científica. Brad<sup>8</sup>, define un agente como "algo que actúa o tiene el poder o autoridad para actuar ... o representar a otro". Russell y Norvig<sup>9</sup>, consideran un agente como una entidad que percibe y actúa sobre un entorno: "algo que percibe estímulos de su ambiente a través de sensores y actúa sobre ese

---

<sup>8</sup> BRADSHAW, J. (ed.) "Software Agents", AAAI Press/ The MIT Press, 1997.

<sup>9</sup> RUSSELL, S., NORVIG, P. "Artificial Intelligence: A Modern Approach", New York, Prentice-Hall, 1995.

ambiente a través de efectores". Franklyn y Graesser<sup>10</sup> extienden su definición de agente, a agente autónomo concluyendo que: "un agente autónomo es un sistema situado dentro un ambiente que percibe estímulos de ese ambiente y actúa sobre él, en el tiempo, para llevar a cabo su propia agenda y así causar lo que él percibirá en el futuro". Teniendo en cuenta estas definiciones, se pueden caracterizar distintos agentes de acuerdo a los atributos que posean (y que van a definir su comportamiento)<sup>11</sup> para resolver un determinado problema.

En la mayoría de las ocasiones, los agentes no son desarrollados de forma independiente sino como entidades que constituyen un sistema. A este sistema se le denomina multi-agente<sup>12</sup>. En este caso los agentes deben o pueden interactuar entre ellos. Las interacciones más habituales como son informar o consultar a otros agentes permiten a los agentes "hablar" entre ellos, tener en cuenta lo que realiza cada uno de ellos y razonar acerca del papel jugado por los diferentes agentes que constituyen el sistema. La comunicación entre agentes se realiza por medio de un lenguaje de comunicación de agentes (ACL –Agent Communication Language).

Un agente va a ser caracterizado por una serie de calificativos, que vienen a denotar ciertas propiedades a cumplir por el agente. Esto lleva a plantear otra definición bastante aceptada de agente donde se emplean tres calificativos que el autor considera básicos. Esta definición ve a un agente como un sistema de computación capaz de actuar de forma autónoma y flexible en un entorno<sup>13</sup>, entendiéndose por flexible que sea:

- Reactivo, el agente es capaz de responder a cambios en el entorno en que se encuentra situado.
- Pro-activo, a su vez el agente debe ser capaz de intentar cumplir sus propios planes u objetivos.
- Social, debe de poder comunicarse con otros agentes mediante algún tipo de lenguaje de comunicación de agentes.

---

<sup>10</sup> FRANKLIN, S., GRAESSER, A. "Is it an Agent or Just a Program? A Taxonomy for Autonomous Agents". In Proc. Of the 3th Int. Workshop on Agent Theories, Architectures, and Languages. New York: Springer-Verlag, 1996.

<sup>11</sup> BOTTI, V., CARRASCOSA, C., JULIAN, V., SOLER, J. The ARTIS Agent Architecture: Modelling Agents in Hard Real-Time Environments. Proceedings of the MAAMAW'99. Lecture Notes In Computer Science, vol. 1647. Springer-Verlag (pag. 63-76), Valencia 1999. ISBN 3-540-66281-2.

<sup>12</sup> HUHNS, M., SINGH, M. P.: Readings in Agents. Readings in Agents. Chapter 1, 1 - 24 1998.

<sup>13</sup> WOOLDRIDGE, M. and JENNINGS, N. R.: Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10(2):115–152, 1995.

Algunas de las características que en la literatura se suelen atribuir a los agentes en mayor o menor grado para resolver problemas particulares y que han sido descritos por autores tales como Franklin y Graesser<sup>14</sup>, y Nwana<sup>15</sup>, éstas son:

- Continuidad Temporal: se considera un agente un proceso sin fin, ejecutándose continuamente y desarrollando su función.
- Autonomía: un agente es completamente autónomo si es capaz de actuar basándose en su experiencia. El agente es capaz de adaptarse aunque el entorno cambie severamente. Por otra parte, una definición menos estricta de autonomía sería cuando el agente percibe el entorno.
- Sociabilidad: este atributo permite a un agente comunicar con otros agentes o incluso con otras entidades.
- Racionalidad: el agente siempre realiza «lo correcto» a partir de los datos que percibe del entorno.
- Reactividad: un agente actúa como resultado de cambios en su entorno. En este caso, un agente percibe el entorno y esos cambios dirigen el comportamiento del agente.
- Pro-actividad: un agente es pro-activo cuando es capaz de controlar sus propios objetivos a pesar de cambios en el entorno.
- Adaptabilidad: está relacionado con el aprendizaje que un agente es capaz de realizar y si puede cambiar su comportamiento basándose en ese aprendizaje.
- Movilidad: capacidad de un agente de trasladarse a través de una red telemática.
- Veracidad: asunción de que un agente no comunica información falsa a propósito.
- Benevolencia: asunción de que un agente está dispuesto a ayudar a otros agentes si esto no entra en conflicto con sus propios objetivos.

No existe un consenso sobre el grado de importancia de cada una de estas propiedades para un agente. Sin embargo, se puede afirmar que estas

---

<sup>14</sup> FRANKLIN, S., GRAESSER, A. Op.cit.

<sup>15</sup> Nwana, H. S.: Software Agents: An Overview. Intelligent Systems Research. AA&T, 1996.



propiedades son las que distinguen a los agentes de los demás programas. Según hemos visto en la definición de Wooldridge<sup>16</sup>, las características de autonomía, reactividad, pro-actividad y sociabilidad son las características básicas.

### **2.3 INGENIERÍA DEL SOFTWARE ORIENTADA A AGENTES (AOSE)**

La Ingeniería del Software Orientada a Agentes es una de las más recientes contribuciones al campo de la Ingeniería del Software. Esto ha generado grandes beneficios, particularmente por la capacidad que tienen los agentes para realizar representaciones en alto nivel de las abstracciones de las entidades activas de un sistema de software.

El propósito principal de la Ingeniería del Software Orientada a Agentes, es crear metodologías y herramientas que permitan el desarrollo y mantenimiento de software basado en agentes. Adicionalmente el software puede ser flexible, fácil de usar, escalable y de alta calidad.

Las metodologías orientadas a agentes, se han planteado como extensión de metodologías orientadas a objetos y metodologías de ingeniería del conocimiento.

La programación orientada a agentes (AOP) puede ser vista como una extensión de la programación orientada a objetos (OOP). En la OOP la entidad principal son los objetos. Un objeto es una combinación lógica de estructuras de datos y sus correspondientes métodos. Los objetos son frecuentemente usados como abstracciones de entidades pasivas en el mundo real. Los agentes son vistos como sucesores de los objetos y como abstracciones de entidades activas. En las metodologías orientadas a objetos son comúnmente empleadas tres vistas para analizar un sistema, las cuales pueden ser interesantes también para analizar un agente: una vista estática, para describir las estructuras de los agentes/objetos y sus relaciones; una vista dinámica para describir interacciones entre objetos agentes y una vista funcional para describir el flujo de datos entre métodos/tareas de los objetos agentes.

También algunas técnicas utilizadas para identificar objetos son propicias para ser aplicadas en la identificación de agentes. Concretamente son de especial interés, las técnicas de casos de uso y tarjetas de Clase-Responsabilidad-Colaboración.

Además de las similitudes de los agentes con los objetos, los primeros soportan también estructuras para la representación de estados mentales como creencias, intenciones, deseos, acuerdos. Adicionalmente los agentes soportan interacciones de alto nivel, (usando lenguajes de comunicación de agentes) que suelen ser

---

<sup>16</sup> WOOLDRIDGE, M. and JENNINGS, N. R. Op.cit.

modeladas como el intercambio de un conjunto predeterminado de actos del habla, como contraposición a los mensajes ad-hoc frecuentemente usados entre objetos. Ejemplos de tales lenguajes son FIPA, ACL, y KQML.

Otra importante diferencia entre AOP y OOP, es que los objetos son controlados desde afuera, en oposición a los agentes que tienen autonomía, y no pueden ser directamente controlados desde afuera. En otras palabras los agentes tienen la posibilidad de decidir ejecutar o no determinadas acciones según la información recibida.

El proceso de Ingeniería de Software Orientado a Agentes AOSE<sup>17</sup> sigue los métodos de diseño Orientado a Objetos como extensión y consiste de tres partes, que son:

- Análisis del Sistema Orientado a Agentes: en esta etapa, se identifican a los agentes que exhiben el comportamiento deseado del sistema. Cada agente tiene su conjunto propio único de capacidades que lo caracterizan.
- Diseño Orientado a Agentes: se define la integración de los agentes identificados en la etapa anterior y los roles de los mismos. Un aspecto importante de los agentes en un sistema multi-agente es que usualmente éstos están organizados en jerarquías. Este modelo de organización también se define en la etapa de diseño.
- Programación Orientada a Objetos: Se elige una arquitectura apropiada de agentes, y los agentes se implementan de acuerdo a la especificación de las dos etapas anteriores.

Por otro lado, debido a que los agentes tienen características cognitivas, las metodologías de ingeniería del conocimiento, pueden proporcionar una base para modelar sistemas multiagente, proporcionando las técnicas de modelado de la base del conocimiento de los agentes. El proceso de adquisición de conocimiento de un agente solo es abordado por estas metodologías y su extensión, puede aprovechar la experiencia adquirida, reutilizar bibliotecas de métodos de resolución de problemas y ontologías así como las herramientas desarrolladas por dichas metodologías.

Problemas planteados en la metodología de ingeniería del conocimiento, aparecen también en el diseño de sistemas multiagentes: adquisición de conocimiento, modelado de conocimiento, representación y reutilización. Sin embargo estas metodologías conciben un sistema basado en conocimiento centralizado y no

---

<sup>17</sup> GROSSE, A., "Position Paper", <http://www.AgentLink.org> DAI Lab. 1995.

abordan los aspectos distribuidos o sociales de los agentes, ni su conducta emprendedora, dirigida por los objetivos.

## **2.4 INGENIERÍA DEL CONOCIMIENTO**

Una de las ramas más importantes de la Inteligencia Artificial es aquella orientada a desarrollar sistemas que repliquen las capacidades humanas de resolución de problemas en dominios específicos. Estos sistemas se han llamado tradicionalmente Sistemas Expertos, aunque actualmente se los conoce como Sistemas Basados en Conocimiento, y han dado lugar a la Ingeniería del Conocimiento.

La ingeniería del conocimiento es aquella disciplina moderna que hace parte de la Inteligencia Artificial que ayuda a construir aplicaciones y sistemas orientados al aprendizaje, apoyándonos en metodologías instruccionales y en tecnología de computación y de telecomunicaciones, intentando representar el conocimiento y razonamiento humanos.

El objetivo de la ingeniería del conocimiento consiste en analizar los métodos de aprendizaje y aplicarlos sobre computadores de forma que sea posible desarrollar un sistema capaz de aprender por si mismo. Los diferentes métodos de aprendizaje son estudiados con el fin de obtener algoritmos que proporcionen a los ordenadores esta capacidad. Las posibles aplicaciones de este campo son innumerables, dado que mediante un conjunto de pautas establecidas, se puede conseguir la resolución de problemas de cualquier tipo.

## **2.5 METODOLOGÍAS ORIENTADAS A AGENTE**

En esta sección se describirán algunas metodologías desde los enfoques de extensión de metodologías orientadas a objetos, a fin de incluir aspectos relevantes de los agentes; y de extensión de metodologías de ingeniería del conocimiento (KE, Knowledge Engineering), tratando de aprovechar las técnicas de la KE para modelar características cognitivas de los agentes:

### **2.5.1 Análisis y diseño OA de Burmesister.**

Define tres modelos para analizar un sistema de agentes: el modelo de agente que contiene los agentes y su estructura interna (creencias, planes, estructuras, etc.); el modelo de organización, que describe las relaciones entre los agentes (herencia y papeles en la organización); y el modelo de cooperación, que describe las interacciones entre los agentes.

**Modelo de agente:** se realiza la identificación de los agentes y su entorno empleando una extensión de las tarjetas CRC para incluir creencias, motivaciones planes y atributos de cooperación.

**Modelo de organización:** propone la identificación de los papeles de cada agente y la elaboración de una jerarquía de herencia y relaciones de agentes empleando OMT.

**Modelo de cooperación:** se identifican las cooperaciones y los participantes en las mismas, analizando los tipos de mensajes intercambiados y los protocolos empleados.

### **2.5.2 Técnica de modelado de agentes para sistemas de agentes de BDI.**

Define dos vistas: externa e interna, para modelar los agentes BDI (Belief, Desire, Intention) Creencia, Deseo e Intención.

La vista externa, descompone al sistema en agentes y define sus interacciones. Para ello se desarrollan dos modelos, el modelo de agente, que describe las relaciones jerárquicas entre clases de agentes y las relaciones entre agentes concretos; y el modelo de interacción que describe las responsabilidades, servicios e interacciones entre los agentes y los sistemas externos.

El proceso de desarrollo de la vista externa comienza con la identificación de los papeles (funcionales, organizativos, etc.) del dominio de la aplicación para identificar los agentes y organizarlos en una jerarquía de clases de agentes (el modelo de clases de agentes). Este modelo se representa con una notación tipo OMT (Object Modeling Technique). Posteriormente se identifican las responsabilidades asociadas a cada papel y los servicios proporcionados y empleados para cumplir con estas responsabilidades e interacciones realizadas en los servicios. Para cada servicio, se identifican las interacciones necesarias y el contenido de información de las mismas. Esta información se recoge en el modelo de ejemplares de agentes.

La vista interna modela cada clase de agente BDI a través de tres modelos: El modelo de creencias, que describe las creencias sobre el entorno, el modelo de objetivos, que describe los objetivos que un agente adopta o los eventos a los que responde; y el modelo de planificación, que describe los planes que un agente puede emplear para alcanzar sus objetivos.

El desarrollo de la vista interna, inicia con el análisis de los diferentes planes para alcanzar el objetivo. Los planes para realizar un evento o alcanzar un objetivo se

describen utilizando una notación gráfica similar a la de los diagramas de estado de Harel, extendiendo la notación para indicar fallos en un plan.

### **2.5.3 Método basado en escenarios multiagente (MASB).**

Propone el desarrollo de MAS en el campo del trabajo cooperativo. La fase de análisis consta de las siguientes actividades:

Descripción de escenarios: identificación con lenguaje natural de los principales papeles desempeñados por agentes de software, agentes humanos y objetos de entorno, y descripción de los escenarios típicos.

Descripción funcional de los papeles: descripción de los papeles de los agentes empleando diagramas de conducta, que describen el proceso desempeñado, la información empleada, y las interacciones con otros agentes.

Modelado conceptual de los datos y del mundo: modelado del conocimiento, datos empleados por cada agente y objetos del mundo a través de diagramas de entidad relación u orientados a objeto, y diagramas de ciclo de vida de las entidades.

Modelo de la interacción sistema-usuario: simulación y definición de las interfaces de interacción hombre-máquina en cada escenario.

El diseño consta de las siguientes actividades:

Descripción de los escenarios y de la arquitectura del MAS: selección de los escenarios que van a ser implementados y de los papeles jugados por los agentes en estos escenarios.

Modelado de objetos: refina el modelado del análisis, añadiendo jerarquías de herencia, atributos y métodos.

Modelado de agentes: especificación de las estructuras de creencias, partiendo de los elementos definidos en el paso de modelado conceptual de los datos en el análisis. Propone una notación gráfica para describir el proceso de decisión de un agente, relacionando creencias, planes, objetivos e interacciones, así como para describir los pasos de cada plan.

Finalmente se propone, aunque no se desarrolla, el modelado de las conversaciones y la validación global del sistema.

#### 2.5.4 Metodología orientada a agentes para modelado de empresas.

Propone la combinación de metodologías orientadas a objetos OOSE (Object Oriented Software Engineering), metodología de modelado de empresas IDEF (Integration DEfinition for Function modelling) y CISMOSA (Computer Integrated Manufacturing Open System Architecture).

Los modelos propuestos son:

**Modelo de funciones:** describe las funciones (entradas, salidas, mecanismos y control) empleando diagramas IDEF0 que incluyen la selección de los métodos posible dependiendo de la entrada y el control.

**Modelo de casos de uso:** describe los actores involucrados en cada función, empleando notación de casos de uso OOSE.

**Modelo dinámico:** Este modelo se menciona para analizar las interacciones entre objetos. Los casos de uso se presentan en diagramas de trazas de eventos.

**Sistema orientado a agentes:** se compone de las siguientes fases:

- Identificación de agentes: los actores de casos de uso se identifican como agentes. Las funciones principales de un agente son sus objetivos y las posibilidades descritas en los diagramas IDEF0.
- Protocolos de coordinación: se describen en diagramas de estados.
- Invocación de planes: se definen diagramas de secuencias que extienden los diagramas de trazas de eventos para incluir condiciones que indiquen cuando un plan es invocado.
- Creencias, sensores y actuadores: las entradas de las funciones deberían ser modeladas como creencias u obtenidas de los objetos mediante sensores, y los objetivos alcanzados deben ser modelados como cambios a las creencias o modificaciones a través de los actuadores.

#### 2.5.5 Metodología CommonKADS.

CommonKADS<sup>18</sup> es una metodología diseñada para el desarrollo de sistemas basados en conocimiento (SBC) de forma análoga a los métodos empleados en

---

<sup>18</sup> R. DE HOOG, R. MARTIL, B. WIELINGA, R. TAYLOR, C. Bright, and W. van de Velde. The CommonKADS model set. ESPRIT Project P5248 KADS-II/M1/DM..1b/UvA/018/5.0, University of Amsterdam, Lloyd's Register, Touche Ross Management Consultants & Free University of Brussels, Dec. 1993.

Ingeniería del Software. El desarrollo de esta metodología ha sido financiado por la Comunidad Europea entre 1983 y 1994 a través de varios proyectos.

La metodología CommonKADS sigue una aproximación al desarrollo de SBC como la construcción de un número de modelos interrelacionados que capturan los principales rasgos del sistema y de su entorno.

El proceso de desarrollo de SBC consiste en tramitar un conjunto de "plantillas" de los modelos.

Asociados a estas plantillas, CommonKADS define "estados" de los modelos que caracterizan hitos en el desarrollo de cada modelo. Estos estados permiten la gestión del proyecto, cuyo desarrollo se realiza de una forma cíclica dirigida por riesgos.

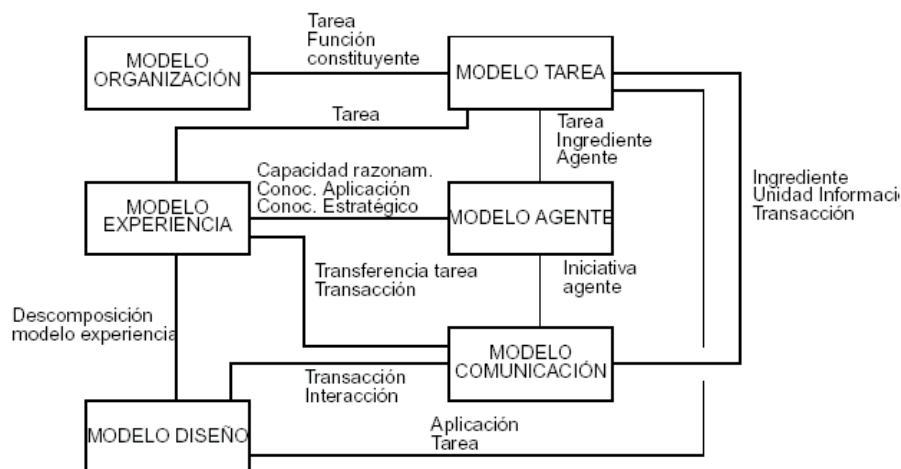
Hay seis modelos definidos en CommonKADS:

**Modelo de la Organización (OM):** es una herramienta para analizar la organización en que el SBC va a ser introducido.

**Modelo de Tarea (TM):** describe a un nivel general las tareas que son realizadas o serán realizadas en el entorno organizativo en que se propone instalar el SBC y proporciona el marco para la distribución de tareas entre los agentes.

**Modelo de Agente (AM):** un agente es un ejecutor de una tarea. Puede ser humano, software o cualquier otra entidad capaz de realizar una tarea. Este modelo describe las capacidades y características de los agentes.

Figura 3. El conjunto de modelos de CommonKADS.



**Modelo de Comunicaciones (CM):** detalla el intercambio de información entre los diferentes agentes involucrados en la ejecución de las tareas descritas en el modelo de tarea.

**Modelo de la Experiencia (EM):** este es el corazón de la metodología CommonKADS y modela el conocimiento de resolución de problemas empleado por un agente para realizar una tarea. El Modelo de la Experiencia distingue entre el conocimiento de la aplicación y el conocimiento de resolución del problema. El conocimiento de la aplicación se divide en tres subniveles: nivel del dominio (conocimiento declarativo sobre el dominio), nivel de inferencia (una biblioteca de estructuras genéricas de inferencia) y nivel de tarea (orden de las inferencias).

**Modelo de Diseño (DM):** mientras que los otros cinco modelos tratan del análisis del SBC, este modelo se utiliza para describir la arquitectura y el diseño técnico del SBC como paso previo a su implementación.

### 2.5.6 Metodología CoMoMAS.

Metodología basado en el enfoque de Ingeniería de Conocimiento, CoMoMAS (Contributed to Knowledge Acquisition and Modelling in a Multi-Agent Framework)<sup>19</sup> propone una extensión de CommonKADS para modelar sistemas multiagente, a través del desarrollo de los siguientes modelos:

**Modelo de agente:** es el modelo central de la metodología y define la arquitectura del agente y el conocimiento del mismo, que se clasifica en social, cooperativo, cognitivo, reactivo y de control.

**Modelo de la experiencia:** describe las competencias cognitivas y reactivas del agente. Distingue entre el conocimiento de tareas, de resolución de problemas y reactivo. El conocimiento de las tareas contiene la descomposición de las tareas, descrita en el modelo de tareas. El conocimiento de resolución de problemas describe los métodos de resolución de problemas (Problem Solving Methods, PSM) y las estrategias para seleccionarlos. El conocimiento reactivo describe los procedimientos para responder a un estímulo.

**Modelo de tareas:** describe la descomposición de las tareas indicando si son resolubles por el usuario o por un agente.

---

<sup>19</sup> GLASSER, N.. Contribution to Knowledge Modelling in a Multi-Agent Framework (the CoMoMAS Approach). PhD thesis, L'Université Henry Poincaré, Nancy I, France, November 1996.



**Modelo de cooperación:** describe la cooperación entre varios agentes. Se descompone en métodos de resolución de conflictos, (métodos de negociación) y conocimiento de cooperación (primitivas, protocolos y terminología de interacción).

**Modelo del sistema:** representa la organización de los agentes y describe la arquitectura del sistema multiagente y sus agentes.

**Modelo de diseño:** Define como pasar de los modelos previos a código ejecutable. Recoge los requisitos funcionales y no funcionales de la aplicación, así como las guías de diseño (plataforma y lenguajes de implementación) y la historia del diseño.

## 2.6 METODOLOGÍA MAS-COMMONKADS

En el anterior numeral se plantean algunas de las metodologías orientadas a agentes más conocidas y que han dado lugar a posteriores estudios y a la conformación de nuevas metodologías que dan solución a los posibles problemas y vacíos que se pudieron presentar a la hora de su aplicación.

Basada en los principios de Ingeniería de Conocimiento, la metodología MAS-CommonKADS<sup>20</sup> extiende los modelos definidos en CommonKADS, añadiendo técnicas de las metodología orientadas a objetos tales como OMT (Object Modelling Technique), OOSE, (Object Oriented Software Engineering) y RDD (Responsibility Driving Design) y de ingeniería de protocolos para describir los protocolos entre agentes, tales como SDL (Specification and Descripción Lenguaje) y MSC96 (Message Sequense Chart) . La metodología comienza con una fase de concepción que emplea la técnica de casos de uso descritos con las notación OOSE y formalizados con MSC (Message Sequense Chart). Define los modelos descritos a continuación para el análisis y el diseño, que son desarrollados siguiendo un ciclo de vida espiral. Para cada modelo se definen sus constituyentes (entidades que son modeladas) y las relaciones entre los constituyentes. Asociado a cada constituyente se define una plantilla textual para describirlo. El estado de desarrollo de cada constituyente (vacío, identificado, descrito o validado) facilita la gestión del proyecto.

### 2.6.1 Modelos de MAS-CommonKADS.

**Modelo de agente:** describe las características de los agentes, su nombre, capacidades de razonamiento, sensores y actuadores, servicios, objetivos, etc. Este modelo incluye una plantilla textual para describir los Agentes y varias

---

<sup>20</sup> IGLESIAS FERNÁNDEZ, Carlos Ángel. Definición de una metodología para el desarrollo de sistemas multiagente MAS-CommonKADS. Enero 1998.

técnicas para identificar los agentes tales como el análisis de los actores de la fase de conceptualización, análisis sintáctico del enunciado del problema, heurísticos para identificar agentes, reutilización de componentes (agentes) desarrollados previamente o empleo de tarjetas CRC, que han sido adaptadas al desarrollo orientado a agentes.

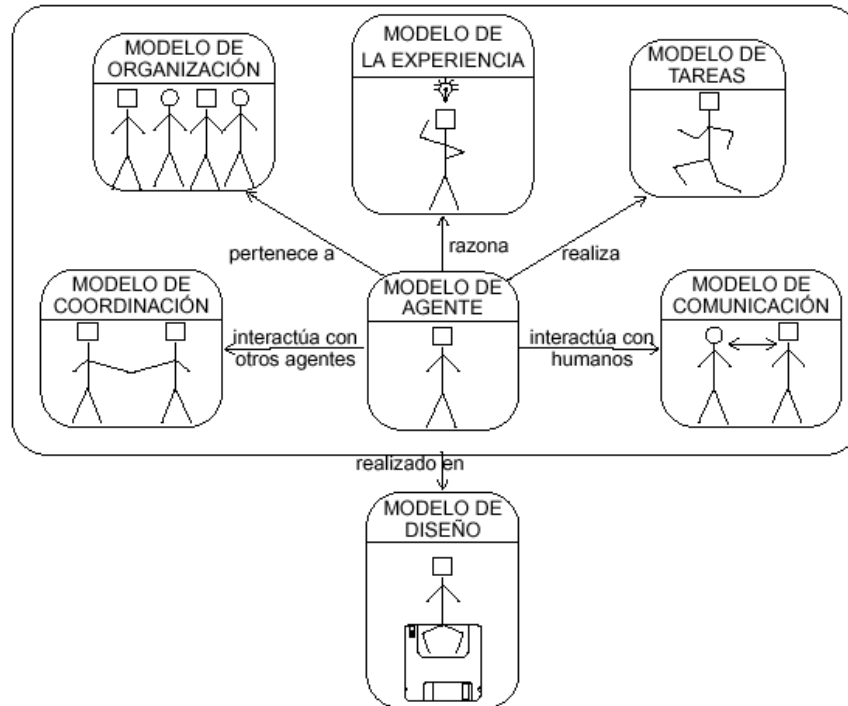
**Modelo de tareas:** describe las tareas (objetivos) realizadas por los agentes y su descomposición, empleando plantillas textuales y diagramas para su desarrollo.

**Modelo de experiencia:** describe el conocimiento que necesitan los agentes para poder conseguir sus objetivos. Se desarrollan varios ejemplares de este modelo para modelar inferencias sobre el dominio, razonamientos sobre el propio agente y sobre otros agentes. Se distingue entre métodos de resolución de problemas autónomos que descomponen un objetivo en subobjetivos hasta llegar a objetivos directamente realizables y métodos de resolución de problemas cooperativos que descomponen un objetivo en objetivos que realiza el agente en cooperación con otros agentes.

**Modelo de organización:** describe tanto la organización en que va a ser introducido el sistema multiagente para estudiar la viabilidad del mismo como la organización de la sociedad multiagente. La descripción de la organización multiagente emplea una extensión de la notación gráfica del modelo de objetos OMT, y describe la jerarquía de agentes, su relación con el entorno y su estructura. Esta extensión define un nuevo símbolo para diferenciar los agentes de los objetos del entorno, que consta de una zona para describir los atributos externos del agente (creencias, objetivos, etc.) y otra para describir los atributos externos (censores, servicios, etc.). La relación de herencia entre agentes para un atributo se define como la unión de los valores de las clases precedentes, pudiendo indicar que se desee no heredar los valores de un atributo.

**Modelo de coordinación:** describe las interacciones de los agentes, los protocolos empleados y las capacidades necesarias para realizar estas interacciones. El desarrollo de este método consta de dos hitos: la definición de los canales de comunicación entre los agentes, que permite el desarrollo de un primer prototipo, y el análisis de las interacciones para determinar cuales de ellas son más complejas y requieren un protocolo de coordinación. Su desarrollo parte de la descripción de los escenarios prototípicos entre los agentes con notación MSC, en cuyos mensajes se especifica el acto de habla realizado y los datos intercambiados. El conjunto de interacciones permitidas entre agentes se recoge en un diagrama de flujo de eventos. El modelado del procesamiento de los mensajes se realiza con diagramas de estados de SDL.

Figura 4. Modelos metodología MAS-CommonKADS.



**Modelo de comunicación:** describe las interacciones hombre-máquina y los aspectos relevantes para desarrollar las interfaces de usuario.

**Modelo de diseño:** recoge los requisitos de los modelos previos y determina como se llevan a cabo. Se subdivide en tres submodelos:

- Diseño de la red: define los requisitos de la infraestructura de la red de los agentes que son proporcionados por agentes de red. Se distinguen tres tipos de facilidades: *facilidades de red*, tales como servicio de nombrado de agentes, páginas blancas y amarillas, protocolos de envío de mensajes disponibles, seguridad de red, contabilidad, etc.; *facilidades de conocimiento*, tales como servidores de ontologías y métodos de resolución de problemas, traductores entre lenguajes de representación del conocimiento, etc.; y *facilidades de coordinación*, que indican los protocolos y primitivas de comunicación consensurados en la red, servidores de protocolos, facilidades de gestión de grupos, detección de malas conductas en el uso de recursos comunes, etc.
- Diseño de los agentes: determina la arquitectura de agente más adecuada para cada agente y descompone cada agente en módulos, que recogen las funciones desempeñadas por cada agente de los modelos de análisis. Por

último se establece una correspondencia entre los módulos y la arquitectura seleccionada.

- **Diseño de la plataforma:** se selecciona y justifica el software y hardware empleando en el desarrollo del sistema multiagente.

### **2.6.2 Fases de desarrollo de MAS-CommonKADS.**

El modelo de ciclo de vida para el desarrollo de sistemas multiagente (con CommonKADS) sigue las siguientes fases:

**Conceptuación:** tarea de extracción o adquisición de conocimiento para obtener una primera descripción del problema y la determinación de los casos de uso que pueden ayudar a entender los requisitos informales y a probar el sistema.

**Análisis:** determinación de los requisitos de nuestro sistema partiendo del enunciado del problema. Durante esta fase se desarrollan los siguientes modelos: organización, tareas, agente, comunicación, coordinación y experiencia.

**Diseño:** determinación de cómo los requisitos de la fase de análisis pueden ser logrados mediante el desarrollo del modelo de diseño. Se determinan las arquitecturas tanto de la red multiagente como de cada agente.

**Codificación y prueba de cada agente:** utilización de herramientas de desarrollo orientadas a agentes.

**Integración:** el sistema completo es probado.

**Operación y mantenimiento:** puesta en funcionamiento y detección de posibles fallas e implementación permanente de actualización.

## **2.7 RECUPERACIÓN DE INFORMACIÓN**

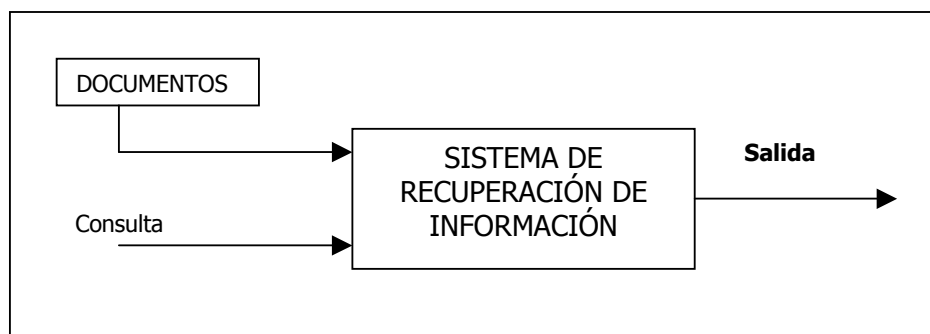
La propuesta se centra en la información gráfica como objeto de trabajo de recuperación de forma genérica asumiendo documento como un elemento multimedia como texto, fotografías, ilustraciones, gráficas, vídeo animado, audio, etc. Recuperar información dispersa en medios masivos, involucra una variedad de capacidades para los mecanismos que los soportan. Los factores principales que se consideran en las decisiones de diseño de estos sistemas, incluyen: la eficiencia, confiabilidad, adecuación, precisión y procesamiento automático de los datos, para responder de buena forma a los requerimientos de los usuarios.

Para lograr una definición más o menos precisa sobre Sistemas de recuperación de Información (SRI) se hace necesario de sus principios en los años cuarenta con Rijsbergen<sup>21</sup> que expone: existe una gran cantidad de información a la que se debe acceder de manera precisa y rápida; y que se complementa con Salton y McGill<sup>22</sup>: la R.I. trata de la representación, almacenamiento, organización y acceso de ítem de información, que inicialmente puede ser cualquier tipo de objeto, que para el caso de estudio se trata de imágenes.

A partir de estas definiciones y para efectos del presente trabajo se define a un SRI como un aplicativo que implementa estas tareas utilizando el procesamiento de un computador para entregar al usuario resultados satisfactorios. Para que esto sea posible el almacenamiento de los documentos se hace mediante representación del mismo en forma de palabras clave o términos de indexación: un conjunto de palabras que resume el contenido del documento, que para éste caso corresponde a rasgos y características que represente la imagen.

El objetivo de la recuperación de información es, dada una necesidad de información y un conjunto de documentos, ordenar los documentos de más a menos relevantes para esa necesidad y presentarlos al usuario. Para ilustrar el problema nos ayudamos de la figura siguiente (adaptada de Rijsbergen<sup>23</sup>).

Figura 5. Sistema de recuperación de información.



En la Figura 5 se ha supuesto que el sistema de recuperación de información es una caja negra que acepta documentos y consultas, y que obtiene una salida, que es el conjunto de documentos que satisfacen la consulta. El problema principal en este sistema es obtener representaciones homogéneas de documentos y consultas,

---

<sup>21</sup> C. J. Van Rijsbergen. Information retrieval. Second Edition. Butter Worths, Lon-don (U.K.), 1979.

<sup>22</sup> SALTON, G. y McGill, M. J.. Introduction to Modern Information Retrieval. McGraw-Hill, Inc., 1983.

<sup>23</sup> C. J. Van Rijsbergen. Op.cit.

y procesar convenientemente esas representaciones para obtener la salida. De otro lado, también es muy importante la evaluación de la salida, para determinar si ésta coincide con las necesidades informativas del usuario.

En el proceso de recuperación de información se suelen distinguir las siguientes etapas:

- Obtener representación de los documentos. Generalmente los documentos se presentan utilizando un conjunto más o menos grande de términos índice. La elección de dichos términos es el proceso más complicado.
- Identificar la necesidad informativa del usuario. Se trata de obtener la representación de esa necesidad, y plasmarla formalmente en una consulta acorde con el sistema de recuperación.
- Búsqueda de documentos que satisfagan la consulta. Consiste en comparar las representaciones de documentos y la representación de la necesidad informativa para seleccionar los documentos pertinentes.
- Obtención de resultados y presentación al usuario.
- Evaluación de los resultados por parte del usuario.

Con el incremento del número de documentos en formato electrónico, se hace necesario contar con herramientas informáticas adecuadas para la recuperación de documentos. Las técnicas manuales han demostrado ser ineficaces, pues básicamente consisten en la elaboración manual de una descripción del contenido temático de cada uno de los documentos, siendo éste un trabajo costoso en tiempo, que además tiene abundante inconsistencia.

Actualmente los computadores hacen posible representar un documento utilizando su contenido completo. Es lo que se denomina representación a texto completo del documento. Efectivamente es la forma más completa de representar un documento, pero implica un costo computacional muy alto para colecciones grandes de documentos.

En estos sistemas cada documento se puede representar por todas sus palabras, tanto nombres, como verbos, adjetivos, adverbios, etc. A pesar de ello, no todos los términos poseen la misma utilidad para describir el contenido de un documento. De hecho, hay términos más importantes que otros, pero no es tarea fácil decidir la importancia de cada término. Por ejemplo, desde el punto de vista de la recuperación de información existen palabras casi vacías de contenido

semántico, como los artículos, preposiciones o conjunciones, que parecen poco útiles en el proceso. Sin embargo, algunos estudios actuales en el campo de la Lingüística indican que el estudio de los artículos incorpora un nivel semántico adicional al nombre que acompañan, y por tanto quizás debieran tenerse en cuenta.

Independientemente de ello, tampoco es útil para las tareas de recuperación de información aquellos términos que se repiten con mucha frecuencia en toda la colección de documentos, pues son términos poco discriminatorios en relación con una consulta dada. En conclusión, en estos sistemas no sólo se persigue encontrar aquellos términos que mejor representen a los documentos, sino además aquellos que permitan diferenciar unos respecto de otros.

Por otra parte, un tercer tipo de sistemas permiten realizar búsquedas conocidas como en texto libre. En éstos se realizan búsqueda de subcadenas, sobre el texto almacenado de todo el documento. Evidentemente, el costo de almacenamiento y computacional en la búsqueda es muy elevado, y además, muchas veces se muestran incapaces de resolver dos problemas básicos: la sinonimia (relación de semejanza de significados entre determinadas palabras) y la polisemia (capacidad que tiene una sola palabra para expresar muy distintos significados). Por ello, la investigación en recuperación de información busca diseñar sistemas que acepten consultas en lenguaje natural y proporcionen documentos adecuados a tales consultas, ordenados según algún criterio del sistema, de acuerdo a las características de los documentos y a las necesidades informativas expresadas por el usuario en su consulta. Uno de los modelos más conocido y difundido para el sistema de recuperación es el llamado modelo vectorial, que se estudiará seguidamente.

### **2.7.1 Modelos de recuperación de información.**

Existe una gran cantidad de modelos de recuperación basados en tecnologías muy diferentes, sin embargo en esta sección se relacionan sólo los tres más utilizados<sup>24</sup>: el booleano, el del espacio vectorial y el probabilístico, asumiendo como modelo de estudio y aplicación el modelo vectorial.

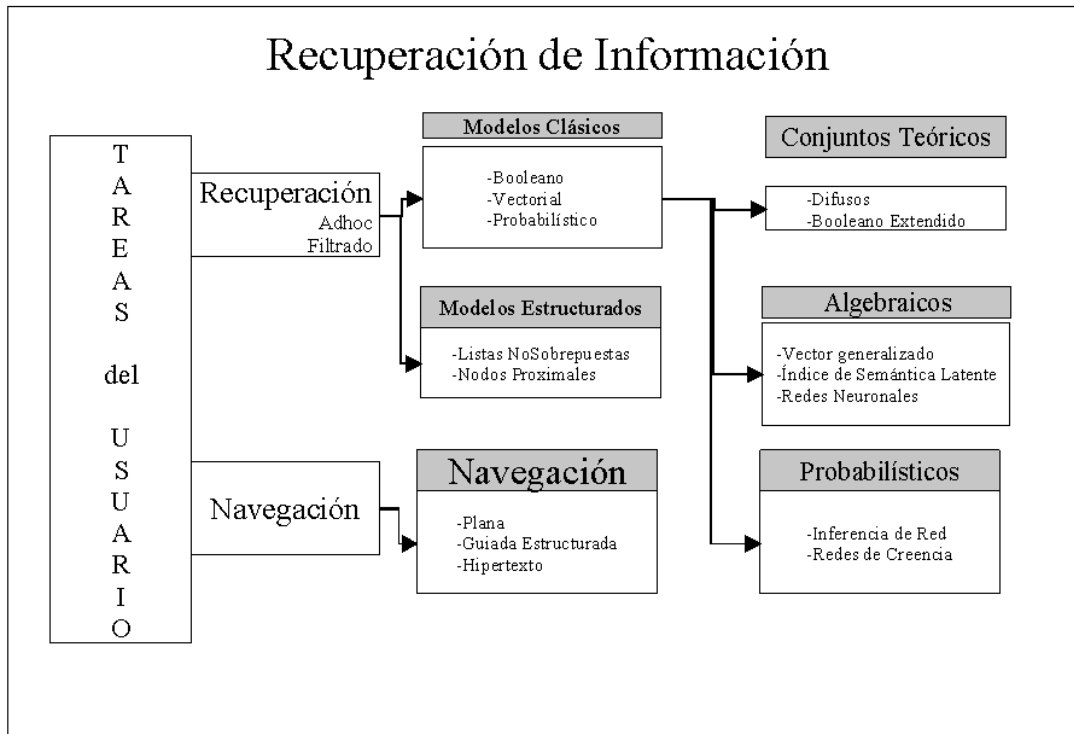
El modelo booleano está basado en la teoría de conjuntos y en el álgebra booleana. Su marco está compuesto por los documentos representados como conjuntos, las consultas, como expresiones booleanas y las operaciones existentes para tratar conjuntos. Los pesos de los términos en los documentos son binarios: 0 indica ausencia, y 1 presencia. Así, en este modelo, dada una consulta al

---

<sup>24</sup> BAEZA-YATES, R. y RIBEIRO-NETO, B. Modern Information Retrieval. Addison-Wesley, Essex, 1999.

sistema, se va evaluando la expresión booleana mediante la realización de las operaciones anteriores con los conjuntos formados por los documentos donde aparece cada término de la consulta. El conjunto de documentos resultante está compuesto por todos aquellos que hacen verdad la consulta booleana.

Figura 6. Modelos de recuperación de información.



En el modelo del espacio vectorial el marco está compuesto por el espacio vectorial representando en él los documentos, las consultas y las operaciones algebraicas sobre los vectores de dicho espacio. Concretamente, la función que obtiene la similitud de un documento con respecto a una consulta se basa en la medida del coseno, la cual devuelve el coseno del ángulo que forman ambos vectores en el espacio vectorial. Así, cuando ambos vectores son exactamente el mismo, el ángulo que forman es de cero grados y su coseno es de uno. Por el contrario, cuando el ángulo que forman es de noventa grados (los vectores no coinciden en ningún término), el coseno será cero. El resto de posibilidades indicarán una correspondencia parcial entre el vector documento y el consulta, ofreciendo así una gradación en los valores de relevancia, de modo que cuanto más cercanos sean los vectores del espacio, más similares serán éstos.



El marco del modelo probabilístico está compuesto por conjuntos de variables, operaciones con probabilidades y el teorema de Bayes. Todos los modelos de recuperación probabilísticos están basados en el principio de ordenación por probabilidad. Este principio asegura que el rendimiento óptimo de la recuperación se consigue ordenando los documentos según sus probabilidades de ser juzgados relevantes con respecto a una consulta, siendo estas probabilidades calculadas de la forma más precisa posible a partir de la información disponible. Así, el objetivo primordial es pasar por calcular la probabilidad de relevancia dados una consulta y un documento.

De acuerdo a lo planteado y a la decisión de tomar como referencia el modelo vectorial es necesario profundizar en su estructura:

El modelo vectorial fue definido por Salton<sup>25</sup> hace ya bastantes años, y es ampliamente usado en operaciones de RI, así como también en operaciones de categorización automática, filtrado de información, etc. En el modelo vectorial se intenta recoger la relación de cada documento  $D_i$ , de una colección de  $N$  documentos, con el conjunto de las  $m$  características de la colección. Formalmente un documento puede considerarse como un vector que expresa la relación del documento con cada una de esas características.

$$D_i \rightarrow \vec{d}_i = (c_{i1}, c_{i2}, \dots, c_{im})$$

Es decir, ese vector identifica en qué grado el documento  $D_i$  satisface cada una de las  $m$  características. En ese vector,  $c_{ik}$  es un valor numérico que expresa en qué grado el documento  $D_i$  posee la característica  $k$ . El concepto 'característica' suele concretarse en la ocurrencia de determinadas palabras o términos en el documento, aunque nada impide tomar en consideración otros aspectos.

Si se consideran los términos como características definitorias del documento, el proceso que debe seguir el sistema pasa primero por seleccionar aquellos términos útiles que permitan discriminar unos documentos de otros. En este punto debemos señalar que no todas las palabras contribuyen con la misma importancia en la caracterización del documento. Desde el punto de vista de la recuperación de información existen palabras casi vacías de contenido semántico, como los artículos, preposiciones o conjunciones, que son poco útiles en el proceso.

Pero también son poco importantes aquellas palabras que por su frecuencia de aparición en toda la colección de documentos pierden su poder de discriminación. En RI todas ellas forman parte del conjunto de palabras vacías (stops words en

---

<sup>25</sup> SALTON, G. Automatic Information Organization and Retrieval. McGraw-Hill, N.Y., 1968.

inglés), que se eliminan en el proceso de indexación. Además de la eliminación de palabras vacías, en el proceso se pueden incluir aplicaciones léxicas como lematización o extracción de raíces, etiquetado de términos, detección de unidades multipalabra, etc.

Una vez seleccionado el conjunto de términos caracterizadores de la colección de documentos, es necesario obtener el valor de cada elemento del vector del documento. El caso más simple es utilizar una aproximación binaria, de forma que si en el documento  $D_i$  aparece el término  $k$ , el valor  $c_{ik}$  sería 1, y en caso contrario sería 0.

No obstante, una palabra puede aparecer más de una vez en el mismo documento, y además, unas palabras pueden considerarse con más peso, esto es, más significativas que otras, de forma que el valor numérico de cada uno de los componentes del vector obedece normalmente a cálculos más sofisticados que la simple asignación binaria. De otro lado, también es importante normalizar los vectores para no privilegiar documentos largos frente a otros documentos menos extensos.

$$\bar{d}_i = \frac{1}{\sqrt{\sum_{j=1}^m w_{ij}^2}} (w_{i1}, w_{i2}, \dots, w_{im})$$

Se han propuesto diversos métodos para calcular el peso de cada término en el vector documento Salton y McGill, 1983<sup>26</sup>; Salton y Buckley, 1988<sup>27</sup>; Harman, 1992<sup>28</sup>, pero en general, para estimarlos se parte de dos ideas en cierto sentido contrapuestas: si un término aparece mucho en un documento, entonces es importante para caracterizar ese documento. Pero si aparece en muchos documentos de la colección, no es beneficioso para distinguir un documento de los demás, dado su escaso poder discriminatorio, resultando poco útil para la recuperación.

Para determinar la capacidad de representación de un término para un documento dado se computa el número de veces que aparece en dicho documento, obteniéndose la frecuencia del término en el documento,  $tf$  (*term frequency*).

---

<sup>26</sup> SALTON, G. y MCGILL, M. J. Op.cit.

<sup>27</sup> SALTON, G.; BUCKLEY, C. Term-Wighting Approaches in Automatic Text Retrieval. En: Information Processing and Management, 24(5), 513-523, 1988.

<sup>28</sup> HARMAN, D. Ranking Algorithms. //Frakes, W.B.; Baeza-Yates, R. Information retrieval: Data Structures and Algorithms. Prentice-Hall, Englewood Cliffs (NJ), pp. 363-392, 1992.

Por otra parte, si la frecuencia de un término en toda la colección de documentos es extremadamente alta, se opta por eliminarlo del conjunto de términos de la colección (pertenece al conjunto de palabras vacías). Podría decirse que la capacidad de recuperación de un término es inversamente proporcional a su frecuencia en la colección de documentos. Esto es lo que se conoce como *idf* (*inverse document frequency*).

Así, para calcular el peso de cada elemento del vector que representa al documento se tiene en cuenta la frecuencia inversa del término en la colección, combinándola de alguna forma con la frecuencia del término dentro de cada documento. Normalmente se utiliza para ello el producto simple<sup>29</sup>.

$$w_{ij} = tf_i \cdot idf_j$$

Salton y Buckley experimentaron con más de 200 sistemas de cálculo de pesos, pero uno de los más utilizados viene dado por la siguiente ecuación, que expresa el peso del término *j* en el documento *i*.

$$w_{ij} = tf_{ij} \cdot \log \frac{N}{df_j}$$

donde *df<sub>j</sub>* es el número de documentos en que aparece el término *j*, y *N* el número de documentos de la colección.

El proceso realizado para los documentos también puede aplicarse a las consultas. Efectivamente, una consulta, *Q*, realizada en lenguaje natural está formada por términos, y, por tanto, puede verse como un documento más, seguramente bastante breve, aunque no siempre. Así pues, el mecanismo de obtención de pesos también se aplica a las consultas, para de esta manera poder disponer de representaciones homogéneas de consultas y documentos, que posibiliten obtener el grado de similitud entre ambas representaciones.

El vector representante de la consulta está formado por un vector de igual número de elementos que los vectores de los documentos. Cada elemento de ese vector expresa el grado en que cada uno de los términos de la colección representa las necesidades informativas de la persona que hace la consulta.

$$Q \rightarrow \vec{q} = \frac{1}{\sqrt{\sum_{j=1}^m p_j^2}} (p_1, p_2, \dots, p_m)$$

---

<sup>29</sup> Ibid.

La resolución de la consulta consiste en un proceso de establecer el grado de semejanza entre el vector consulta y el vector de cada uno de los documentos. Para una consulta determinada, cada documento arrojará un grado de similitud determinado; aquellos cuyo grado de similitud sea más elevado se ajustarán mejor a las necesidades expresadas en la consulta, desde el punto de vista del sistema de recuperación de información. No obstante, es el usuario el que debe decidir la relevancia de los documentos recuperados, siendo ésta una característica totalmente subjetiva del mismo.

El modo más simple de calcular la similitud entre una consulta y un documento, utilizando el modelo vectorial, es realizar el producto escalar de los vectores que los representan. En esa ecuación se incluye la normalización de los vectores, a fin de obviar distorsiones producidas por los diferentes tamaños de los documentos. El índice de similitud más utilizado es el coseno del ángulo formado por ambos vectores. Para una consulta  $Q$ , el índice de similitud con un documento  $D_i$  es:

$$SIMIL(Q, D_i) = \frac{\sum_{j=1}^m p_j d_{ij}}{\sqrt{\sum_{j=1}^m p_j^2 \sum_{j=1}^m d_{ij}^2}}$$

Hay otros métodos propuestos para calcular la similitud; un cuadro con los más importantes puede encontrarse en Salton y McGill, 1983. Los resultados de la computación del índice de similitud entre la consulta y todos los documentos permite ordenar los resultados en orden decreciente. De esta manera se le ofrecen al usuario primero los documentos que el sistema de recuperación considera más similares con la consulta, y que pueden coincidir, o no, con lo esperado por el usuario. La relevancia es la medida subjetiva que el usuario tiene para determinar si los resultados, y en qué grado, son adecuados a sus necesidades informativas.

### **2.7.2 Evaluación de la recuperación.**

Un SRI, una vez ordena los documentos según la similitud a la consulta y entrega esta lista al usuario, habría acabado su función (salvo que se desee aplicar algún tipo de método para mejorar la consulta), pero si se está utilizando un sistema experimental, se pondrá en marcha un último módulo que determinará el rendimiento del mismo. Este rendimiento puede calcularse en función del elemento que se evalúe: desde el tiempo que tardaría el sistema en responder después de efectuar la consulta, hasta la cantidad de material relevante, pasando por cuestiones como la forma de presentar las salidas, el espacio que necesita

para llevar a cabo las acciones correspondientes, o el esfuerzo requerido por el usuario para formular la consulta<sup>30</sup>.

## 2.8 PROCESADO DE TEXTO

Una vez analizado el modelo vectorial y la evaluación de los sistemas de recuperación, es necesario detenerse un momento a estudiar el análisis de los elementos que componen cada vector que representa a documentos y consultas. Hemos dicho que cada elemento del vector expresa en qué grado el documento o consulta posee una característica de la colección. En general, el concepto característica se concreta en la aparición de determinadas palabras o términos en el documento. Ahora bien, el proceso pasa por seleccionar aquellos términos que mejor representen los documentos y que mejor discriminen unos documentos respecto de otros.

Como ya indicamos en un apartado anterior, no todas las palabras de un documento son igual de representativas del contenido del mismo. Por ejemplo, en lenguaje escrito las palabras que actúan gramaticalmente como nombres suelen tener más significado que el resto de palabras, aunque no siempre es así. Así pues, la fase inicial del proceso consiste en analizar el texto del documento para determinar qué términos pueden utilizarse como términos índice. Esta fase se denomina habitualmente como preprocesado de texto de la colección de documentos, y suele llevar la sucesión de los siguientes pasos<sup>31</sup>:

- Análisis léxico del texto con el objetivo de determinar el tratamiento que se realizará sobre números, guiones, signos de puntuación, tratamiento de mayúsculas y/o minúsculas, nombres propios, etc.
- Eliminación de palabras vacías con el objetivo de reducir el número de términos con valores muy pocos discriminatorios para la recuperación. Es una forma de delimitar el número de términos índice como términos índice.
- Aplicación de lematización sobre los términos resultantes para eliminar variaciones morfo-sintácticas y obtener términos lematizados.
- Selección de los términos (o grupos de términos) que serán términos índice. Normalmente se realiza sobre la naturaleza sintáctica del término, pues, como ya se ha mencionado, los términos que actúan gramaticalmente como nombres suelen poseer un mayor contenido semántico que verbos, adjetivos o adverbios.

---

<sup>30</sup> C. J. Van Rijsbergen. Op.cit.

<sup>31</sup> BAEZA-YATES, R. y RIBEIRO-NETO, B. Op.cit.

- Construcción de tesauros, es decir, diccionarios de términos asociados, que permitan expandir la consulta con los términos relacionados.

## **2.9 BASES DE DATOS DE IMÁGENES**

La gestión de archivos consiste en una serie de pasos interrelacionados, diseñados para asegurar la fácil identificación, organización, acceso y mantenimiento de los archivos. Para el caso de información gráfica le aporta un componente adicional, ya que otorga atributos que permiten una aplicación contextualizada y una herramienta valiosa en el campo editorial.

Las teorías en torno a las bases de datos de imágenes incluye primero puntualizar el concepto de base de datos<sup>32</sup> y su aplicación para cuando se trata de formatos gráficos, almacenamiento, sus tipos y requerimientos físicos y lógicos<sup>33</sup>, además de conocer formatos gráficos, entornos y sistemas de información necesarios<sup>34</sup> para la toma de decisiones al momento de determinar y delimitar condiciones mínimas y necesidades para el implementación del sistema.

Los sistemas de asignación de nombres de archivos y directorios por omisión son rara vez óptimas para una colección específica. Las decisiones sensatas acerca de los archivos y los directorios pueden ayudar a minimizar el caos, en especial de colecciones muy grandes. Hasta cierto punto, la naturaleza del material que se está almacenando sugerirá los procesos de organización. Las series con frecuencia se dividen en volúmenes y números, las monografías tienen número de páginas, las colecciones de manuscritos o de fotografías tienen números de carpeta o de acceso, etc. En la mayoría de los casos, algún aspecto de estos principios de organización física pueden traducirse a organización de sistemas de archivos.

Los materiales gráficos aportan a la organización de los archivos índices que permiten categorizar por área, que para el presente proyecto permite contextualizar sus características en torno a su aplicación en el campo editorial.

### **2.9.1 Definición de bases de datos.**

Una base de datos es un conjunto de elementos de datos que se describe a sí mismo, con relaciones entre esos elementos, que presenta una interfaz uniforme

---

<sup>32</sup> JONSON, James L. Bases de Datos. Modelos, lenguajes, diseño. OXFORD UNIVERSITY PRESS. Mexico 1997.

<sup>33</sup> CORNELL UNIVERSITY. Library Gateway. Llevando de la Teoría a la Práctica. Tutorial de Digitalización de Imágenes.

<sup>34</sup> LÓPEZ YEPES, Alfonso. Revista. Cuadernos de Documentación Multimedia Número 3. Fototecas Digitales en Prensa: formatos gráficos, entornos y sistemas de información. Junio 1994.

de servicio. Un sistema de administración de bases de datos (DBMS) es un producto de software que presta soporte al almacenamiento confiable de la base de datos, pone en marcha las estructuras para mantener relaciones y restricciones, y ofrece servicios de almacenamiento y recuperación a usuarios; más funciones se ocupan de otras tareas, como son el acceso simultáneo, seguridad, respaldo y recuperación (lectura) de datos.

Muchas de las primeras iniciativas digitales confiaban en la programación personalizada para manejar grandes colecciones de archivos de imágenes. Las rutinas para el procesamiento por lotes, la organización y entrega de archivos, se escribían utilizando lenguajes de texto de alto nivel, como por ejemplo Pert y Tcl. Hoy en día existen muchos productos ya armados que pueden simplificar radicalmente el proceso de manejar una gran cantidad de archivos de imágenes. Sin embargo, aún el sistema más simple requiere algún tipo de personalización.

Las bases de datos de imágenes varían significativamente en cuanto a la facilidad de uso y al nivel de funcionalidad. Realizan un seguimiento de sus archivos, proporcionan funciones de búsqueda y recuperación, suministran una interfaz de acceso, controlan el nivel y tipo de uso, y proporcionan algo de seguridad al controlar quién tiene acceso a qué. Ninguna herramienta tiene la posibilidad de satisfacer todas sus necesidades, in incluso el conjunto de herramientas elegido más cuidadosamente necesita ser reevaluado en forma regular para determinar si aún sigue siendo la mejor selección.

Los criterios generales para evaluar las bases de datos de imágenes incluyen los siguientes:

- Objetivo para el cual se creó la colección digital.
- Tamaño y tasa de crecimiento de la colección digital.
- Complejidad y volatilidad de los metadatos complementarios.
- Nivel de demanda y de rendimiento esperado.
- Infraestructura técnica existente, incluyendo disponibilidad del personal de sistemas capacitado.
- Costos.

Categorías básicas de los sistemas de bases de datos: Una evaluación cuidadosa de los sistemas de gestión de imagen, incluyendo las virtudes y defectos de cada

tipo y ejemplo de aplicación, son tratados por Peter Hirtle<sup>35</sup> sugiere las siguientes categorías fundamentales:

Las bases de datos de escritorio comunes son medianamente económicas y simples de usar, pero limitadas en cuanto a tamaño y funcionalidad.

Las aplicaciones de bases de datos cliente-servidor son más costosas y sofisticadas que las bases de datos de escritorio, pero también son más difíciles de utilizar y mantener.

Los sistemas especializados de gestión de imágenes pueden ofrecer una completa solución ya armada, con estructuras de datos predefinidas, pero son más costosas y menos flexibles en términos de personalización y compatibilidad.

Más sistemas de biblioteca están teniendo capacidades de soportar imágenes. Aquellos que ya las tienen, ofrecen buen enlace entre los registros de catálogo existentes y las imágenes digitales, pero sufren de falta de estandarización y de una preferencia por el enlace a nivel de artículo. El personal de los sistemas de biblioteca puede no estar preparado para tomar la carga adicional de manejar grandes colecciones de imágenes. Sin embargo, ésta es un área de desarrollo intensivo. En los próximos años veremos más sistemas de biblioteca que contengan bases de datos de imágenes.

### **2.9.2 Almacenamiento.**

El almacenamiento masivo es una de las tecnologías informáticas más competitivas y que más rápido avanza. Como resultado, el almacenamiento está en una curva de disminución precio / rendimiento que compite con los microprocesadores.

Los criterios generales para la evaluación incluyen:

- Velocidad (lectura / escritura, transferencia de datos);
- Capacidad;
- Fiabilidad (estabilidad, redundancia);
- Estandarización;
- Costo;

---

<sup>35</sup> HIRTLE, Peter. Moving Theory into Practice: Digital Imaging for Libraries and Archives.



- Aptitud para la tarea.

Los criterios para seleccionar el almacenamiento masivo apropiado para una iniciativa de digitalización parecen bastante sencillos. La clave es evitar los cuellos de botella, por lo cual el objetivo es maximizar la velocidad con la cual se accede a los datos y se los transfiere. Con las colecciones digitales que están en constante crecimiento, también es la capacidad, y con frecuencia los dispositivos con la mayor capacidad están entre los más rápidos, y proporcionan el menor costo por unidad de almacenamiento.

### **2.9.3 Tipos de almacenamiento.**

Las tecnologías de almacenamiento masivo se pueden clasificar de distintas maneras. El sistema de almacenamiento subyacente (magnético, óptico o magenetoóptico), el tipo de unidad (fija o removible), y el material del medio (cinta, disco rígido, disco flexible) en forma conjunta definen las características de cada tecnología.

Todas las tendencias básicas en el almacenamiento masivo, distintas de la proliferación confusa de nuevas tecnologías, son positivas para las imágenes digitales. La velocidad y la capacidad continúan aumentando, así como también la fiabilidad. Los costos han ido disminuyendo a la mitad cada año, desde por lo menos 1988. Ya que el costo de almacenamiento en disco magnético ha disminuido relativamente respecto de otros tipos de almacenamiento, ha dominado el mercado del almacenamiento primario.

### **2.9.4 Necesidades de almacenamiento**

Las necesidades están determinadas en capacidades de almacenamiento, el costo de almacenamiento y las consideraciones de hardware y de software.

Las necesidades de almacenamiento pueden ser estimados por medio de un simple cálculo:

Almacenamiento total necesario = cantidad de archivos de imágenes x tamaño de archivo promedio x 1,25

Donde 1,25 proporciona muchos otros factores que pueden aumentar las necesidades de almacenamiento como por ejemplo la tecnología de almacenamiento utilizada.

También el costo total de almacenamiento se puede estimar mediante la siguiente fórmula:

Costo total de almacenamiento = almacenamiento total necesario x costo por unidad de almacenamiento.

Esto proporciona una estimación aproximada, dado que sólo incluye costos básicos de unidad y medios. Otros costos relacionados con el almacenamiento incluye estantería y estuches, fuentes adicionales de energía eléctrica en caso de cortes, cables, tarjetas, software de gestión de almacenamiento, etc.

Las bases de datos, por definición, están diseñadas para funcionar en sistemas de escritorio bajo MacOS o Windows. Sin embargo, incluso una colección pequeña puede verse sobrecargada en un sistema de escritorio si demasiados usuarios intentan acceder a la misma en forma simultánea. La mayoría de las aplicaciones de bases de datos más grandes están diseñadas para funcionar en entornos de usuarios múltiples, como Unix, Linux o Windows NT/2000, que se ejecutan en máquinas que ofrecen rápidos procesadores, mucha memoria RAM, rápidos buses de entrada / salida y periféricos, y rápidos dispositivos de almacenamiento.

### **2.9.5 Formatos gráficos.**

Desde que la evolución de la tecnología permitió la introducción de imágenes en una plataforma informática, la ausencia de un estándar ha sido la característica común. Con el continuo avance de la técnica, los sistemas incrementan sus prestaciones y capacidad, lo que se refleja en la aparición de nuevos formatos gráficos. Un ejemplo de lo afirmado es la "reciente" aparición del formato JPEG, basado en esquemas de compresión disipadora o con pérdida de datos.

Sin embargo, a estas alturas del desarrollo, sí existen algunos tipos de archivo que destacan por encima del resto gracias a su capacidad, versatilidad y difusión. Primeramente es necesario hacer una distinción entre los formatos raster o de barrido y el EPSF, formato PostScript desarrollado para ilustraciones e infográficos. Cada uno tiene una finalidad distinta y la manera de trabajar con ellos también es diferente.

Los primeros (de barrido) están basados en mapas de bits, es decir, en la descripción de los píxeles mediante coordenadas, mientras que el segundo está orientado a objeto (se describen los objetos que forman una imagen -líneas, polígonos, curvas, etc.) y su ubicación.

Existen múltiples formatos basados en mapa de bits, pero sólo se relacionan tres: aquellos que pueden ser de verdad operativos en un archivo informatizado que ha de servir imágenes de gran calidad a múltiples finalidades, tanto en color como en blanco y negro (hoy predominante en el mundo de la prensa diaria).

En primer lugar el TIFF (Tagged Image File Format), es el formato quizá más conocido. Soporta imágenes desde 1 a 24 bits por píxel (desde formato por líneas o line art hasta calidad fotográfica) y puede comprimirse.

Otro formato de amplia y creciente difusión es el JPEG, con unas prestaciones en cuanto a reducción del tamaño de los archivos realmente asombrosas. La clave radica en la antes mencionada pérdida de información, pero en cantidades prácticamente tan insignificantes que no suponen un grave inconveniente para su uso.

El último de los formatos es el PHOTO CD, un desarrollo específico de Kodak que no puede ser definido como "formato" a secas, ya que conlleva mucho más: toda una filosofía de la imagen digital. Su reciente aparición en el mercado todavía no ha permitido su asentamiento como estándar, pero su innovadora solución a la fotografía doméstica digital y el aval de tener como respaldo a la primera empresa mundial en fotografía -Kodak- facilitará mucho el proceso en un breve plazo.

Finalmente hay que comentar que algunos formatos gráficos son susceptibles de ser comprimidos (como el TIFF), lo que permite aprovechar mejor el sistema de almacenamiento. Cuando las imágenes cuentan con una profundidad de más de un bit por píxel (sólo para gráficos lineales en blanco y negro), resulta pobre el uso de técnicas de compresión de imágenes binarias estandarizadas por el CCITT (como los algoritmos Grupo III y Grupo IV), y es necesario acudir a otros tipos de formatos como son las técnicas de compresión JPEG o el TIFF comprimido.

Dentro de los posibles esquemas de compresión, existen dos grandes tipos:

Los libres de pérdidas, o los lossless, que alcanzan ratios de 2:1 de media, lo que no soluciona nada cuando es necesario almacenar cientos de grandes ficheros (una imagen en color de 24 bits por píxel puede ocupar con facilidad 25 Mb) en cualquier base de datos gráfica. El funcionamiento de estos sistemas de compresión sin pérdida se fundamenta en la búsqueda de modelos de información o repetitivas cadenas de código dentro de los archivos para sustituirlos por pequeñas referencias. Cuanto más frecuente es el modelo, más testigos puede meter el programa, y cuanto más largo sea el modelo, más espacio se ahorra al sustituirlo con un testigo. Normalmente se utilizan para fotografías en escala de grises, ya que en su codificación se dan grandes niveles de redundancia. Las

imágenes en color, por otro lado, varían enormemente en sus niveles de redundancia, por lo que esta compresión no resulta tan aconsejable.

Por otra parte, los esquemas de compresión con pérdida o compresión disipadora son utilizados sobre todo cuando los originales son analógicos (imágenes, vídeo y sonido), factor que posibilita que la pérdida de información no sea traumática.

Cualquiera de los dos tipos de compresión hace más lenta la lectura, pero la mayoría de las veces el ahorro de espacio justifica con creces los tiempos de espera.

## **2.10 PERFIL**

El profiling es una técnica cuya idea principal es la de crear y gestionar perfiles. Un perfil es un modelo de un objeto (una representación compacta que describe las características más importantes), creado en la memoria del computador, y que es utilizado como representante (del modelo actual) del objeto, en las tareas computacionales. Estas tareas incluyen por ejemplo: comparaciones, almacenamiento, resúmenes y análisis<sup>36</sup>.

Las aplicaciones de profiling más conocidas incluyen la personalización, la gestión de conocimiento y el análisis de datos.

### **2.10.1 ¿Qué es un perfil?**

Perfil es una palabra que viene de las palabras del latín "pro filare", que significan "diseñar los contornos". Una cabeza vista de lado a blanco y negro –una silueta– es uno de los perfiles más conocidos. Proyecta las características más notables de una cabeza en un plano, de forma que ésta aún pueda ser reconocida.

Los perfiles en el mundo de los computadores cumplen la misma tarea: describir las características más vitales del objeto, en un formato compactado, que los computadores entiendan, para que los programas los puedan usar en sus tareas.

Existe un número indeterminado de distintos perfiles, desde el perfil psicológico del comportamiento de un criminal, hasta el perfil del funcionamiento de un programa de computador. En principio, se puede hacer un perfil de todo, y por consiguiente, las características representadas en el perfil dependen de la naturaleza del objeto modelado.

---

<sup>36</sup> P. P. DE LA CRUZ, Rui Alexandre. Perfiles de Usuario: En la senda de la personalización. Informe Técnico. Departamento de Informática y Automática. Universidad de Salamanca. Enero 2003.

En el mundo de los computadores, el objetivo de la mayor parte del profiling, es la parte más compleja e imprevisible de un ambiente informático: el usuario. No se hacen solamente perfiles de los seres humanos como usuarios sino también como clientes. Este tipo de profiling se ha incrementado mucho en los últimos años en las tiendas en línea y en las aplicaciones de gestión de las relaciones con los clientes (Customer Relationship Management – CRM).

Un perfil es una colección de propiedades de un objeto. Cuando se hace un perfil de un usuario en línea, estas propiedades incluyen, por ejemplo, el método de conexión utilizado, el terminal utilizado, patrones de comportamiento e intereses del usuario.

Cuando se hace el perfil de un cliente, las propiedades normalmente preferidas incluyen la edad, la residencia habitual, el nivel del salario, así como las distintas formas de contacto (email, teléfonos - fijo o móvil, etc.), hábitos de compra, etc.

Para los seres humanos el nombre es seguramente una de las propiedades más importantes y también un método de identificación.

También se hacen perfiles de documentos. En este caso las propiedades más importantes suelen ser el nombre del documento, el autor, la fecha de creación y las palabras.

Por otro lado, si se habla del perfil de un producto, las características incluyen, por ejemplo, el productor, precio e información técnica del mismo.

### **2.10.2 Aplicaciones del profiling.**

La gestión de conocimiento y contenidos se basa principalmente en el perfil de documentos. En estos sistemas todos los documentos tienen un perfil que puede ser utilizado para ordenar, filtrar, procurar documentos, y controlar versiones. La idea principal es la de encontrar los documentos adecuados en los enormes, normalmente distribuidos y muchas veces desestructurados, conjuntos de documentos.

Los perfiles de los clientes son utilizados con el mismo propósito en los call centers computarizados. Cuando una llamada es atendida se coloca automáticamente la información del cliente en la pantalla.

Los perfiles son comprensibles por los sistemas de información para que puedan ser utilizados en análisis complejos. Las relaciones entre los perfiles pueden ser examinadas y los perfiles pueden ser agrupados de distintas formas (clusters). Por

ejemplo, en algunos supermercados organizan los productos sobre la base de los perfiles de compra creados a partir de los registros de venta de las cajas registradoras. Los clientes también pueden ser divididos en segmentos, basándose en su perfil de cliente, por ejemplo, para dirigir sus campañas promocionales.

También los agentes de software pueden utilizar los perfiles para automatizar tareas y ayudar los usuarios de forma pro-activa. Por ejemplo, un agente de búsqueda puede investigar redes de ficheros para encontrar aquellos que coincidan con el perfil del usuario.

#### 2.10.2.1 Personalización:

Personalización es el conjunto de procesos en el que un sistema provee servicios basados en la información personal de cada usuario. Los tipos de personalización se dividen en muchas categorías, de las cuales cabe destacar: reconocimiento por nombre, customization y adaptativa.

El reconocimiento por nombre es la forma más básica, en que el usuario es reconocido y saludado al llegar al servicio.

La customization es normalmente llamada "personalización de caja de opción", porque el usuario define los parámetros de funcionamiento del servicio, seleccionando sus preferencias de una lista de cajas de opciones. Este tipo de personalización es frecuentemente usada por los sistemas de hipermedia adaptable, en los que el perfil contiene las opciones seleccionadas.

La personalización de servicios adaptativa, hace una personalización avanzada, seleccionando el contenido de sus presentaciones, de acuerdo con las acciones del usuario anteriormente realizadas en ese servicio. Esta información es procesada y guardada en su perfil de usuario durante la interacción del usuario con el sistema, y posteriormente analizada para adaptar la presentación. Este método de personalización es el más corriente.

La personalización permite entablar una relación de marketing "uno-a-uno" real, para un gran número de usuarios. No divide los usuarios en grupos, sino que selecciona para cada usuario separadamente la presentación, utilizando para ello tecnologías de Inteligencia Artificial (IA), como son: redes neuronales, redes bayesianas, lógica borrosa y razonamiento basado en casos. También se suelen emplear heurísticas basadas en el dominio de aplicación para mejorar el desempeño de los sistemas.

Un componente de personalización debe ser capaz de recomendar documentos, promover productos, aconsejar apropiadamente, etc.

La personalización es cada vez más utilizada como medio de agilizar la entrega de información al usuario, convirtiendo al sistema en más útil y atractivo, de forma que el usuario se sienta estimulado al utilizarlo. Por eso, la personalización se viene convirtiendo en un requisito esencial en los sistemas de hoy en día, y en particular en los sitios web de comercio electrónico.

La personalización en comercio electrónico permite ofertas de productos, ventas promocionales, publicidad, etc. dirigidas individualmente al usuario, tomando en consideración su historial de navegación, compras y otras interacciones con la tienda electrónica.

#### 2.10.2.2 Gestión del conocimiento y análisis de datos:

La Gestión del Conocimiento<sup>37</sup> es el proceso de gestionar explícitamente los activos no materiales. La gestión del conocimiento se ha producido a través de generaciones, desde que los humanos empezaron a contar historias unos a otros. La novedad es que está siendo considerado como una actividad de negocio con el objetivo de obtener beneficios comerciales.

Esto quiere decir que las herramientas para la gestión del conocimiento realmente no gestionan el conocimiento, pero ayudan a capturarlo, almacenarlo y transmitir el material a partir del cual el individuo adquiere el conocimiento.

El conocimiento es una capacidad humana más que una propiedad de objeto inanimado como un libro o un disco. Es una capacidad personal como la habilidad, la experiencia y la inteligencia. Se puede adquirir como resultado de leer, escuchar o ver algo. Lo que se lee, se escribe u oye no es conocimiento, es material a través del cual el conocimiento se transfiere. Esto significa que las herramientas gestoras de conocimiento no manipulan el conocimiento, sino que lo capturan, organizan, almacenan y transmiten materiales fuente del cual alguien puede adquirir el conocimiento.

El conocimiento es el resultado de una transformación personal. La adquisición del conocimiento depende de la interacción entre la persona y el material. Un material que transfiere conocimiento a un científico no lo transmite a un artista. La transformación depende de la relevancia, el contexto y cómo el material está estructurado.

---

<sup>37</sup> ARROLLO MUÑOZ, Ana. Gestión del Conocimiento. Departamento de consultoría tecnológica. Fundación Robotiker. Revista Robotiker.

El motivo de la existencia del gestor de conocimiento es conseguir una herramienta para comunicar, colaborar, y coordinar el conocimiento, la transferencia de habilidades, etc. y que todo esto se traduzca en un aumento de la productividad e innovación.

Lo que se pide a un Gestor de Conocimiento:

- Facilitar la conversión de la información en inteligencia y conocimiento, por ejemplo, representando gráficamente las relaciones de las diferentes fuentes de información.
- Herramienta que contempla otras utilidades y métodos que tenga la compañía. Integrar información obtenida de proveedores de contenidos, herramientas y motores de búsqueda, herramientas de inteligencia de negocio, tecnología de búsqueda por texto, etc.
- Herramienta que integre información desde variedad de formatos electrónicos, papel, y fuentes personales. Bases de datos corporativas, proveedores de contenido, servicios on-line, Internet, news, etc.
- Integración de estas herramientas en los sistemas para que se aproveche la inversión existente.



## 3. ANÁLISIS

### 3.1 INTRODUCCIÓN

El análisis de sistemas es el primer y fundamental paso dado clásicamente hacia la construcción de una aplicación informática y que para efectos del presente trabajo se parte del concepto universal de análisis de un sistema de información, tratando el caso específico de análisis orientado a objetos como referentes para abrir paso al análisis orientado a agentes desde la metodología MAS-CommonKADS.

La definición de WHITTEN<sup>38</sup> define el análisis de sistemas como el estudio de una aplicación del sistema de información y de empresa actual y la definición de las necesidades y las prioridades de usuario para conseguir una aplicación nueva o mejorada, distribuido en fases y actividades que permiten estudiar la viabilidad, comprender el sistema y la definición y establecimiento de prioridades desde las necesidades del usuario.

Desde la ingeniería del software, según PRESSMAN<sup>39</sup>, la tarea del análisis de requisitos, es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del software, inicialmente establecida por el ingeniero de sistemas y refinada durante la planificación temporal del proyecto de software. Se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y se asignan a diferentes elementos del software. El modelo de análisis es la primer representación técnica de un sistema, para el cual existen dos tendencias: análisis estructurado y análisis orientado a objetos.

El análisis estructurado es una actividad de construcción de modelos. Mediante una notación que satisface los principios de análisis operacional, se crean modelos que representan el contenido y flujo de la información; se parte el sistema funcionalmente, y según los distintos comportamientos se establece la esencia de lo que se debe construir. El modelo debe lograr describir lo que requiere el cliente, establecer una base para la creación de un diseño de software, y definir un conjunto de requisitos que se puedan validar una vez que se ha construido el software.

---

<sup>38</sup> WHITTEN, Jeffrey L. Análisis y Diseño de Sistemas de Información. Tercera Edición. Editorial Mc GrawHill. 1996.

<sup>39</sup> PRESSMAN, Roger S. Ingeniería del Software. Cuarta Edición. Editorial McGrawHill. 1997.

Y el análisis orientado a objetos tiene como propósito definir todas las clases (y las relaciones y los comportamientos asociadas con ellas) que son relevantes al problema que se va a resolver. El objetivo es desarrollar una serie de modelos que describan el software al trabajar para satisfacer un conjunto de requisitos definidos por el cliente. El análisis orientado a objetos, como los métodos de análisis convencional, forman un modelo de análisis multiparte para satisfacer este objetivo. El modelo de análisis ilustra información, funcionamiento y comportamiento dentro del contexto de los elementos del modelo de objetos que se basa en objetos y atributos, clases y miembros, un todo y partes.

Según la metodología MAS-CommonKADS, el resultado de la fase de análisis es la especificación del sistema compuesto a través del desarrollo de los modelos de agente, organización, tareas, experiencia, comunicación y coordinación, determinando las condiciones iniciales para describir la arquitectura y el diseño del sistema como paso previo a su implementación, modelos que serán detallados y aplicados en los siguientes numerales.

## **3.2 CONCEPTUACIÓN**

En la fase de conceptualización para la implementación del sistema de recuperación de información gráfica se realiza la identificación y descripción de los actores y la identificación y descripción de los casos de uso. La metodología para identificar los actores del sistema consiste en determinar los actores involucrados en el mismo teniendo en cuenta la descripción del problema.

### **3.2.1 Enunciado del problema.**

La producción de materiales electrónicos o impresos implica además de teorías actualizadas y abiertas y de un modelo metodológico que lo estructure, la incorporación de un componente de gran importancia como es la ilustración, la utilización de imágenes que complementen y haga menos rígido el uso de los materiales para que puedan cumplir su cometido de facilitador y apoyo al conocimiento, trabajo que resulta en extremo complejo en tiempo y con pocas posibilidades de éxito debido a la creciente cantidad de información gráfica existente surgiendo la necesidad de una herramienta que permita recuperar dicha información a partir de los textos utilizados para dicho fin con la posibilidad adicional de enriquecer y mejorar en la medida de su uso la calidad y efectividad de los resultados.

### **3.2.2 Identificación de los actores.**

El sistema interactúa con: usuarios del sistema y de mantenimiento como agentes humanos, y los sistemas de bases de datos y el editor utilizado para la elaboración de los materiales, que son actores de software, que se describen a continuación:

### **3.2.3 Descripción de los actores.**

#### **ACTOR Usuario**

Descripción: usuario experto en edición y elaboración de materiales impresos o electrónicos, responsable del concepto gráfico de dichos materiales.

#### **ACTOR Editor**

Descripción: programa o herramienta que permita incorporar formatos gráficos que corran bajo entornos Windows. Los programas van desde herramientas ofimáticas comerciales como Office, AbiWord, StarOffice, OpenOffice, Works, Word Perfect, etc, hasta herramientas profesionales de edición de textos como Page Maker, así como para herramientas desarrolladoras de páginas Web como Front Page, Dream Weber, etc, y el desarrollo de materiales electrónicos como Director de Macromedia.

#### **ACTOR BDatos**

Descripción: base de datos que permite la identificación, organización, acceso y mantenimiento de los archivos gráficos. Para el caso de información gráfica le aporta un componente adicional, ya que otorga atributos desde el perfil de trabajo y documentación de la misma, que permiten una aplicación contextualizada y una herramienta valiosa en el campo editorial.

#### **ACTOR Mantenimiento**

Descripción: encargado de la administración de la base de datos. Deberá contar con privilegios especiales de acceso y criterios según el tipo de información.

### **3.2.4 Identificación de los casos de uso.**

#### **ACTOR Usuario**

Como primera acción el usuario personaliza la herramienta y determina los perfiles de usuario y de trabajo que va a utilizar para lograr el primer acercamiento al tipo

de imágenes que va a requerir (personalización y perfiles por área de trabajo, por categoría y otros) con lo cual se dispone a ingresar contenidos al editor, entendiéndose por editor al programa que procesa textos y permite la incorporación de formatos gráficos, contenidos que son la base para determinar los atributos que la imagen debe tener, aplicativo que estará dispuesto una vez ejecutado el sistema y determinados los perfiles. Una vez hecha la solicitud, el usuario obtiene un grupo de imágenes preseleccionadas por el sistema, luego de haber procesado el texto y consultado en la base de datos las palabras relevantes en búsqueda de imágenes exitosas, y para las cuales puede aplicar una de las siguientes operaciones:

- Seleccionar una imagen si una salida es considerada exitosa.
- Modificar atributos sobre la documentación de la imagen según criterio de relevancia del usuario.

Cualquier proceso hecho por el usuario de actualización y modificación de los atributos de la documentación de la base de datos permite mejorar la efectividad de futuros resultados con lo que se prevé que a mayor uso de la herramienta mayor precisión se logra.

Si la búsqueda no es exitosa o que sencillamente el usuario no se apoya del texto para la búsqueda, el usuario tiene la posibilidad de asignar atributos de búsqueda directamente permitiendo una de las siguientes operaciones:

- Solicitar imagen según criterios del usuario.
- Cambiar perfiles para ubicar la consulta en un nuevo contexto.
- Asignar atributos directamente al sistema.
- Editar los atributos de la documentación de las imágenes exitosas.

### **ACTOR Editor**

El editor es el escritorio sobre el cual se va a ensamblar los materiales, que permite el ingreso de textos relacionados y seleccionados previamente por un experto en contenidos y con los que se va a solicitar al aplicativo selección de imágenes relacionadas para posteriormente insertarlas y aplicar, de acuerdo al editor, características de tamaño, ubicación y atributos especiales de diseño.

## **ACTOR BDatos**

Contiene el banco de imágenes, sus atributos producto de la definición de posibles perfiles de trabajo y palabras genéricas que la describan, flexible y escalable que permiten clasificaciones para garantizar resultados exitosos.

## **ACTOR Mantenimiento**

La actualización, depuración y alimentación permanente de la BDatos con material de libre distribución o autorizado y su información indexada.

### **3.2.5 Descripción de los casos de uso.**

#### **ACTOR Usuario:**

Caso de uso *AsignarPerfiles*

**Resumen:** el usuario determina los perfiles de trabajo y de usuario.

**Actores:** usuario.

**Precondiciones:** ninguna.

**Descripción:** con la asignación de perfiles se personaliza el aplicativo y delimita a un grupo de posible imágenes que estén relacionadas por área de trabajo o categoría y permite obtener la información básica de usuario con el que se identificará en sus sesiones de trabajo.

**Excepciones:** ninguna.

**Postcondiciones:** el usuario obtiene los perfiles de trabajo que podrían ser modificadas en caso de no obtener del sistema resultado exitoso.

Caso de uso *DeterminarAtributosDeBúsqueda*

**Resumen:** procesamiento de texto.

**Actores:** usuario, editor.

**Precondiciones:** utilizar como idioma predeterminado el español y que el texto esté estructurado por párrafos.

**Descripción:** el usuario ingresa textos sobre el editor previa programación de temas y distribución por parte del experto en contenidos de cada área, o bien manualmente directamente ingresando al aplicativo parámetro de búsqueda.

**Excepciones:** ninguna.

**Postcondiciones:** permite al sistema obtener un grupo de criterios de búsqueda utilizando herramientas de procesamiento de texto para su depuración.

#### Caso de uso *SolicitarImagen*

**Resumen:** con activar el sistema quedan disponible las operaciones necesarias para la recuperación de las imágenes preseleccionadas.

**Actores:** usuario.

**Precondiciones:** definidos perfiles y contar con textos en el editor o alimentado algunos atributos de búsqueda.

**Descripción:** el usuario solicita imágenes preseleccionadas por el sistema a partir de un texto para su inserción en el editor, permitiendo refinar la búsqueda manualmente en caso de no encontrar la imagen esperada.

**Excepciones:** ninguna.

**Postcondiciones:** bajo el criterio del usuario poder seleccionar la imagen correspondiente.

#### Caso de uso *SeleccionarInsertarImagen*

**Resumen:** determinar la relevancia de una imagen.

**Actores:** Usuario, BDatos, Editor.

**Precondiciones:** obtener imágenes exitosas.

**Descripción:** el usuario una vez ubicada la imagen exitosa, a partir de la preseleccionadas por el sistema, permite su inserción en el editor.

**Excepciones:** el banco de imágenes no posea las imágenes esperadas.

**Postcondiciones:** el usuario obtiene una imagen exitosa o le permite una nueva búsqueda cambiando perfiles o atributos manualmente hasta finalizar con éxito la operación.

#### Caso de uso *AgregarAtributos*

**Resumen:** agregar manualmente criterios para nuevas búsquedas.

**Actores:** Usuario, Editor, BDatos, Mantenimiento.

**Precondiciones:** contar con atributos nuevos de una imagen exitosa.

**Descripción:** la posibilidad de agregar atributos de imagen a los ya existentes con base al texto tomado de referencia para ampliar el rango de atributos a buscar sobre la documentación de BDatos y actualizarla.

**Excepciones:** ninguna.

**Postcondiciones:** nuevos atributos de imagen.

#### Caso de uso *EditarPerfiles*

**Resumen:** cambio de los perfiles de trabajo iniciales para nuevas búsquedas.

**Actores:** Usuario.

**Precondiciones:** recuperación no exitosa.

**Descripción:** el usuario puede redefinir los perfiles para ampliar o delimitar el rango de imágenes a buscar una vez se determine por parte del usuario que la operación inicial no fue exitosa.

**Excepciones:** ninguna.

**Postcondiciones:** nuevos criterios de búsqueda.

#### **ACTOR Editor:**

##### Caso de uso *EdiciónTexto*

**Resumen:** área de trabajo que permite la edición de texto y manejo de imágenes.

**Actores:** Usuario, Editor.

**Precondiciones:** editor de texto y que permita la inclusión de formatos gráficos para entornos Windows.

**Descripción:** ingreso de texto sobre el área de trabajo incorporando formatos y herramientas para el procesamiento de texto.

**Excepciones:** ninguna.

**Postcondiciones:** texto base de criterios de búsqueda.

##### Caso de uso *InsertarImágenes*

**Resumen:** incorporación de imágenes al área de trabajo.

**Actores:** Usuario, Editor.

**Precondiciones:** la existencia de imágenes exitosas y que el editor permita insertar imágenes.

**Descripción:** operación propia del editor que permite la incorporación de formatos gráficos y aplicar operaciones básicas de formato y ajustes.

**Excepciones:** ninguna.

**Postcondiciones:** insertar una imagen relacionada al texto de referencia.

#### **ACTOR BDatos:**

##### Caso de uso *AlmacenarImágenes*

**Resumen:** permite la creación del banco de imágenes.

**Actores:** Usuario, Mantenimiento.

**Precondiciones:** modelo de base de datos predeterminada.

**Descripción:** la base de datos de imágenes almacena imágenes y sus atributos para que puedan ser clasificadas.

**Excepciones:** ninguna.

**Postcondiciones:** el usuario contará con un banco de imágenes.

##### Caso de uso *AgregarModificarAtributos*

**Resumen:** atributos a partir asignación de criterios de perfil por área de trabajo, criterios a partir de criterios de categoría y características de cada imagen de la base.

**Actores:** BDatos, Usuario, Mantenimiento.

**Precondiciones:** banco de imágenes.

**Descripción:** conjunto de atributos propios de cada imagen que permitirá clasificarlas por perfiles, categorías y características las preferencias del usuario, aplicando herramientas propias de edición sobre una base de datos como agregar, modificar y suprimir, y que además debe permitir conocer las palabras no relacionadas existentes en su documentación para efectos de agregar a la base de datos.

**Excepciones:** ninguna.

**Postcondiciones:** un banco de imágenes con atributos relacionados.

## **ACTOR Mantenimiento**

Caso de uso *ActualizarBDatos*

**Resumen:** almacenar y asignar atributos a las imágenes.

**Actores:** Usuario, BDatos.

**Precondiciones:** ninguna.

**Descripción:** actualización permanente del banco de imágenes y sus atributos. Los atributos se ingresan automáticamente o de forma manual a partir de la experiencia.

**Excepciones:** casos en que una característica no tenga correspondiente genérica en la BDatos con las búsquedas, para lo cual permanece la documentación ingresada inicialmente hasta que seguramente con la permanente actualización sea modificada.

**Postcondiciones:** ampliación y depuración de la documentación del banco de imágenes.

## **3.3 MODELO DE AGENTE**

### **3.3.1 Identificación de los agentes.**

A partir de la conceptualización y del enunciado del problema se puede determinar inicialmente tres agentes: agente Usuario, agente humano que es el operador del aplicativo, agente BDatos, agente de software encargado de almacenar la información necesaria para el proceso, y Agente Recuperación, agente de software que estará a cargo, a partir de criterios de búsqueda desde el texto de referencia capturado en el editor y de la información suministrada por el usuario al momento de definir sus perfiles, de realizar la recuperación exitosa de las imágenes y la actualización permanente de atributos y criterios de cada imagen y el grado de relevancia de cada uno de ellos sobre la imagen, logrando modificaciones permanente que ubiquen atributos cada vez más cercanos a imágenes exitosas. Para este agente se debe tomar en consideración los pesos que adquieran los atributos según sus coincidencias y los pesos de las imágenes según su relevancia,



así como tareas adicionales para posibles resultados no favorables, y tareas propias del manejo de bases de datos, que favorezcan al sistema en futuros usos.

### 3.3.2 Descripción de los agentes.

Plantillas textuales:

Agente *Usuario*

**Tipo:** agente humano.

**Papel:** suministra información requerida para la búsqueda.

**Posición:** externo.

**Descripción:** representa a los beneficiarios expertos en edición y elaboración de materiales impresos y electrónicos, responsable del concepto gráfico de dichos materiales, agente que a partir de la información suministrada recibe como resultado un conjunto de imágenes a escoger para su quehacer.

**Objetivo:** suministra información de usuario y perfil de trabajo, generar contenidos de referencia para búsquedas, selecciona imágenes exitosas y editar perfiles y atributos de imagen.

**Parámetros de entrada:** información del área de trabajo y sobre objeto de producción del material, además de conocimiento básico sobre el uso de herramientas de edición, del prototipo y concepto gráfico.

**Parámetros de salida:** perfiles de usuario, perfil de trabajo, texto escrito en el editor y/o palabras suministradas en el cuadro de texto (atributos manuales) y concepto sobre nuevos atributos de imagen.

**Servicio:** aporta perfiles, textos base de búsqueda y determina la pertinencia de los resultados de la búsqueda y modifica información adjunta si es necesario.

**Experiencia:** concepto gráfico.

**Comunicación:** Agente BDatos.

**Coordinación:** Agente Recuperación.

Agente *BDatos*

**Tipo:** agente de software.

**Papel:** suministra y almacena toda la información requerida para los demás agentes.

**Posición:** interno.

**Descripción:** contiene el banco de imágenes, sus categorías producto de la definición de posibles perfiles de trabajo y los atributos que la describan, flexible y escalable que permiten clasificaciones para garantizar resultados exitosos.

**Objetivo:** guardar, editar, eliminar y agregar información.

**Parámetros de entrada:** imágenes, atributos, categorías y perfiles.

**Parámetros de salida:** imágenes preseleccionadas y su información relacionada.

**Servicio:** almacenar información.

**Experiencia:** modificación de atributos.

**Comunicación:** Agente Usuario.

**Coordinación:** Agente Recuperación.

#### *Agente Recuperación*

**Tipo:** agente de software.

**Papel:** recuperar imágenes, aprender del uso y optimizar su información.

**Posición:** interno.

**Descripción:** este agente recupera imágenes de la BDatos de acuerdo a sus pesos de relevancia, actualiza sus pesos, de acuerdo a los criterios de búsqueda, permitiendo actualizar automática o o manualmente su información (agregar, modificar o quitar atributos) enriqueciendo y refinando la información inicial.

**Objetivo:** predetermina imágenes exitosas.

**Parámetros de entrada:** información de perfiles y criterios de búsqueda.

**Parámetros de salida:** imágenes recuperadas y sus atributos, reasignación de pesos a atributos e imágenes, que involucra automáticamente los pesos de categorías y de perfiles.

**Servicio:** recuperar información gráfica.

**Experiencia:** imágenes exitosas.

**Comunicación:** Agente BDatos.

**Coordinación:** Agente Usuario.

A partir de la conceptualización, la identificación de los actores y los casos de usos, con la identificación y descripción de los agentes, se permite la identificación y desarrollo de constituyentes de los modelos de tareas, coordinación y comunicación.

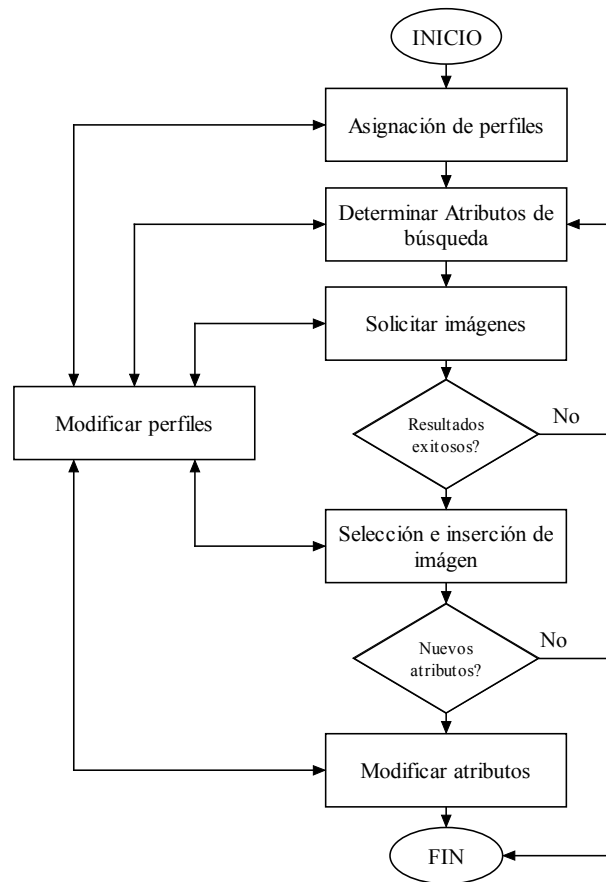
### **3.4 MODELO DE TAREAS**

El modelo de tareas va a permitir la descripción de las actividades relacionadas para alcanzar un objetivo, logrando la descomposición funcional del sistema proporcionando un mejor entendimiento de los procesos internos que realiza.

Para la definición de tareas se presenta en forma textual y aplica las plantillas de definición de tareas para describir las actividades de cada uno de los agentes:

## Agente Usuario

Figura 7. Diagrama de flujos Agente Usuario.



### Tarea: Asignar Perfiles

**Objetivo:** personalizar: asignar perfiles de usuario y de trabajo.

**Descripción:** con la asignación de perfiles se personaliza el aplicativo y delimita a un grupo de posible imágenes que estén relacionadas por área de trabajo según las categorías que la conforme y permite obtener la información básica de usuario con el que se identificará en sus sesiones de trabajo.

**Ingrediente:** perfiles.

**Restricciones:** ninguna.

**Excepciones:** ninguna.

### Tarea: Determinar Atributos de Búsqueda

**Objetivo:** asignar atributos de la imagen a buscar.

**Descripción:** el usuario ingresa textos sobre el editor previa programación de temas y distribución por parte del experto en contenidos de cada área, o bien manualmente ingresando al aplicativo parámetro de búsqueda.

**Ingrediente:** atributos.

**Restricciones:** párrafo o líneas de texto que contengan información relevante para la búsqueda en idioma español.

**Excepciones:** que los atributos suministrados correspondan a ninguna de las imágenes almacenadas.

**Tarea:** Solicitar Imagen

**Objetivo:** requerir la búsqueda de imágenes relevantes.

**Descripción:** el usuario inicia la recuperación a partir de un texto para su inserción en el editor, permitiendo refinar la búsqueda manualmente en caso de no encontrar la imagen esperada.

**Ingrediente:** texto base.

**Restricciones:** ninguna.

**Excepciones:** ninguna.

**Tarea:** Seleccionar e Insertar Imagen

**Objetivo:** determinar la relevancia de las imágenes preseleccionadas por el sistema.

**Descripción:** el usuario una vez ubicada la imagen exitosa, a partir de la preseleccionadas por el sistema, permite la inserción al editor en el punto que se encuentre el cursor.

**Ingrediente:** imagen.

**Restricciones:** según el concepto gráfico del usuario para su selección.

**Excepciones:** resultados no exitosos.

**Tarea:** Agregar Atributos

**Objetivo:** agregar y modificar atributos de imágenes preseleccionadas.

**Descripción:** la posibilidad de agregar atributos de imagen a los ya existentes con base al texto tomado de referencia para ampliar el rango de atributos a buscar sobre la documentación de BDatos y actualizarla.

**Ingrediente:** nuevos atributos.

**Restricciones:** atributos de mayor relevancia.

**Excepciones:** que el atributo relevante ya exista en la información inicial.

**Tarea:** Editar Perfiles.

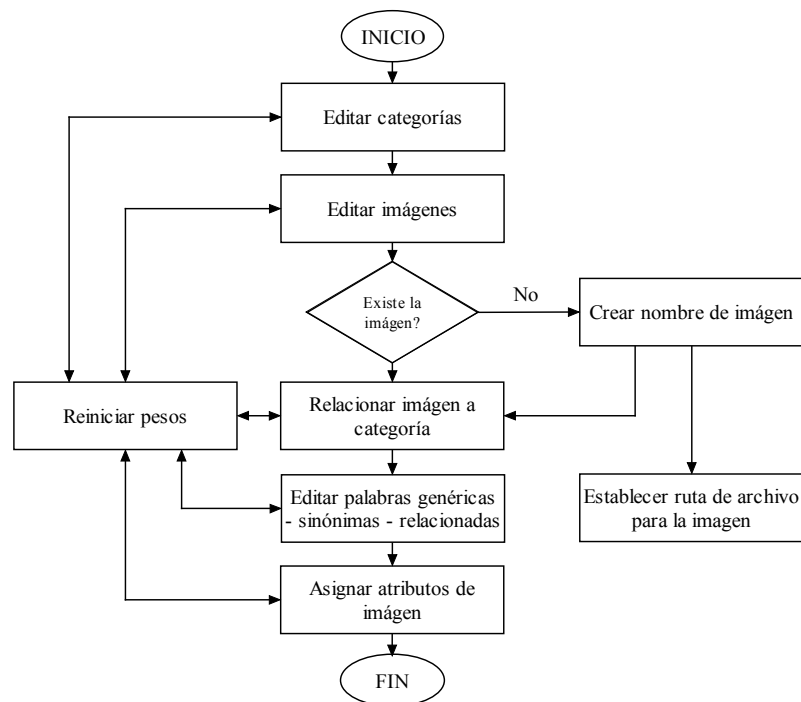
**Objetivo:** modificar la información inicial de los perfiles.

**Descripción:** el usuario puede redefinir los perfiles de trabajo para ampliar o delimitar el rango de imágenes a buscar una vez se determine por parte del usuario que la operación inicial no fue exitosa.

**Ingrediente:** nuevos perfiles.  
**Restricciones:** ninguna.  
**Excepciones:** ninguna.

### Agente Bdatos

Figura 8. Diagrama de flujos Agente Bdatos.



**Tarea:** Editar categorías.

**Objetivo:** crear, modificar o eliminar categorías.

**Descripción:** encargada de la edición de categorías que agrupa las imágenes según sus características.

**Ingrediente:** nombres de categorías.

**Restricciones:** ninguna.

**Excepciones:** ninguna.

**Tarea:** Editar Imágenes

**Objetivo:** crear, modificar o eliminar imágenes.

**Descripción:** en la base de datos se almacena imágenes para que puedan ser clasificadas y manipuladas.

**Ingrediente:** información gráfica e información indexada a cada gráfica.

**Restricciones:** ninguna.

**Excepciones:** ninguna.

**Tarea:** Relacionar imágenes a categorías.

**Objetivo:** determinar la imágenes de cada categoría.

**Descripción:** asigna el grupo de imágenes que corresponde a cada categoría según sus características.

**Ingrediente:** imágenes, categorías.

**Restricciones:** existir imágenes y categorías en la base de datos.

**Excepciones:** ninguna.

**Tarea:** Editar palabras genéricas, sinónimas y palabras relacionadas.

**Objetivo:** crear, modificar o eliminar palabras.

**Descripción:** edición de las palabras genéricas, sinónimas y relacionadas sobre la base de datos.

**Ingrediente:** palabras genéricas, palabras sinónimas y palabras relacionadas.

**Restricciones:** idioma español, sin tilde, en minúsculas, cadenas alfabéticas y sólo sustantivos.

**Excepciones:** ninguna.

**Tarea:** Editar atributos a imágenes.

**Objetivo:** determinar los atributos de cada imagen.

**Descripción:** relaciona palabras genéricas a cada imagen que aportan información sobre su contenido.

**Ingrediente:** imágenes y palabras genéricas.

**Restricciones:** ninguna.

**Excepciones:** ninguna.

**Tarea:** Reiniciar pesos.

**Objetivo:** reiniciar los valores de los pesos de las imágenes, categorías y perfiles de trabajo.

**Descripción:** los pesos de las imágenes, categorías y perfiles de trabajo se van incrementando de acuerdo a los resultados de la recuperación, esta tarea permite reiniciar estos valores con fines demostrativos y comparativos.

**Ingrediente:** imágenes, categorías, perfiles de trabajo y sus pesos.

**Restricciones:** ninguna.

**Excepciones:** ninguna.

## **Agente Recuperación**

**Tarea:** Determinar perfiles de usuario y de trabajo - configuración

**Objetivo:** aplicación de perfiles.

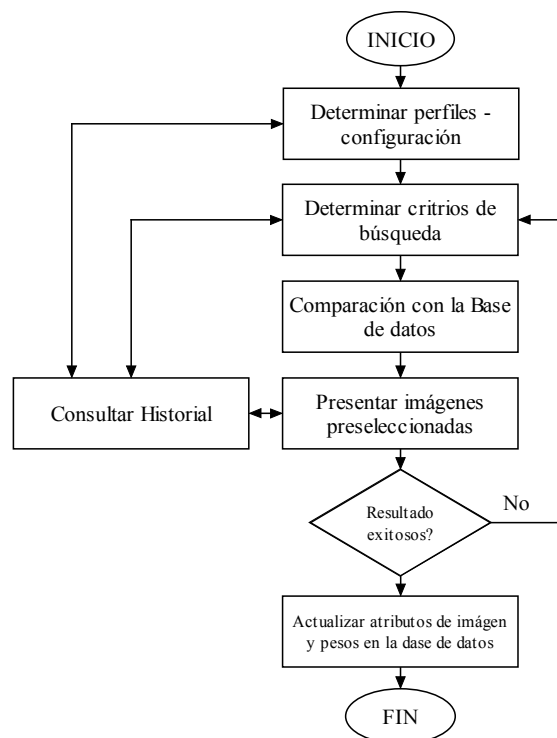
**Descripción:** crear, modificar o eliminar perfiles de usuario y de trabajo o seleccionarlos si ya existen y cumplen las expectativas del usuario.

**Ingrediente:** información del usuario y categorías.

**Restricciones:** que no se repitan usuarios.

**Excepciones:** ninguna.

Figura 9. Diagrama de flujos Agente Recuperación.



**Tarea:** Determinar Criterios de Búsqueda.

**Objetivo:** procesar el texto.

**Descripción:** a partir de la información proveniente de los perfiles y del texto de referencia para la recuperación, se determina un conjunto de palabras como referente de búsqueda sobre los atributos de las imágenes de la base de datos.

**Ingrediente:** texto de referencia.

**Restricciones:** idioma español.

**Excepciones:** ninguna.

**Tarea:** Comparación con la BDatos

**Objetivo:** relacionar los criterios de búsqueda con la información almacenada por coincidencias.

**Descripción:** se basa en la correspondencia que exista entre los criterios de búsqueda y la información almacenada como atributos de las imágenes.

**Ingrediente:** ninguno.

**Restricciones:** el idioma español.

**Excepciones:** que palabras base para la búsqueda no estén incluidas en las palabras genéricas.

**Tarea:** Presentación de Resultados Obtenidos

**Objetivo:** entregar al usuario el resultado de la búsqueda.

**Descripción:** la consulta a la base de datos permite preseleccionar un grupo de imágenes relevantes las cuales son presentadas al usuarios para su selección, que en caso de no ser exitosos los resultados, reiniciaría el proceso a partir de nuevos parámetros de búsqueda.

**Ingrediente:** atributos e imágenes.

**Restricciones:** resultados no exitosos de la búsqueda.

**Excepciones:** ninguna.

**Tarea:** Actualización BDatos

**Objetivo:** actualizar atributos y pesos.

**Descripción:** las imágenes son almacenadas con unos atributos iniciales que serán modificados permanentemente a fin de aumentar el éxito en futuras búsquedas.

**Ingrediente:** categorías, atributos, perfiles y pesos.

**Restricciones:** ninguna.

**Excepciones:** ninguna.

**Tarea:** Consultar Historial.

**Objetivo:** obtener información de las sesiones de trabajo de los usuarios.

**Descripción:** información sobre registro de las sesiones de trabajo realizadas por el usuario.

**Ingrediente:** información de los perfiles, fecha y hora.

**Restricciones:** ninguna.

**Excepciones:** ninguna.

### **3.5 MODELOS COORDINACIÓN, COMUNICACIÓN, ORGANIZACIÓN**

A partir de su definición, el desarrollo de estos modelos se hace de forma aislada, tomando como base los modelos de agente y tareas como únicas dependencias en su construcción, siguiendo las siguientes actividades:

El modelo de coordinación permite la identificación y descripción de las interacciones entre los agentes de un sistema multiagente. El principal objetivo de este modelo es modelar la interacción agente-agente.

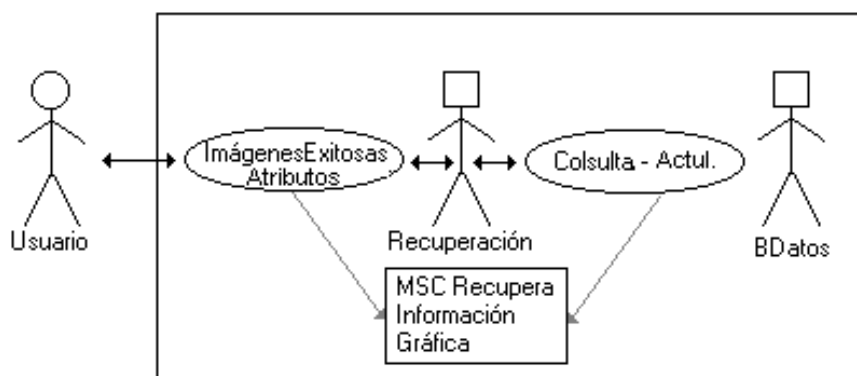


Con el modelo de coordinación se establecen las interacciones entre agentes de software, dejando para el modelo de comunicación, de forma específica, el estudio de las interacciones concretas hombre-máquina y de los factores humanos que deben tenerse en cuenta para el diseño de tales interfaces.

### 3.5.1 Identificación de conversaciones.

La identificación de conversaciones se representa de forma gráfica utilizando "casos de uso internos" para determinar escenarios de conversación entre los agentes humanos y no humanos.

Figura 10. Caso de uso interno.



### 3.5.2 Descripción de las conversaciones.

En el caso de los agentes de software Recuperación y BDatos, se determina la conversación Consulta y Actualización, con la siguiente descripción textual:

**Conversación:** Consulta - Actualización

**Tipo:** suministro y actualización de información.

**Objetivo:** comparar las palabras relevantes provenientes del agente Recuperación con la información indexada para cada imagen, como perfiles y atributos, presentar la información existente en la base de datos asociada a la imagen y permite su actualización (modificación).

**Agentes:** Recuperación, BDatos.

**Iniciador:** Recuperación.

**Servicio:** comparación de datos.

**Descripción:** a partir de las palabras relevantes obtenidas en el procesamiento del texto, se realiza una comparación con la información indexada a las imágenes, que permitirá preseleccionar un grupo de imágenes, además de que según estos resultados se provea de nueva

información a la base de datos como actualización de pesos, nuevos atributos.

**Precondición:** grupo de palabras relevantes.

**Postcondición:** imágenes asociadas al grupo de palabras procesadas, nueva información para la base de datos.

**Condición-terminación:** la existencia de palabras coincidentes y la modificación de la información asociada a las imágenes en la base de datos.

**Tiempo-ejecución:** entre 10 y 60 segundos.

El modelo de comunicación, entonces, se encarga de escribir la conversación ImágenesExitosas - Atributos entre los agentes Usuario y Recuperación así:

**Conversación:** ImágenesExitosas - Atributos

**Tipo:** obtención de información gráfica con sus atributos y la generación de nuevos atributos.

**Objetivo:** obtener imágenes exitosas y atributos relacionados y aporta nuevos atributos a las imágenes.

**Agentes:** Usuario, Recuperación.

**Iniciador:** Usuario.

**Servicio:** imágenes preseleccionadas y atributos.

**Descripción:** a partir de los perfiles de usuario y de trabajo y de un conjunto de palabras, el agente Recuperación procesa el texto y determina un grupo de palabras relevantes que serán utilizadas para consultar en la base de datos, para finalmente entregar al usuario un grupo de imágenes con sus atributos para ser utilizadas por el usuario en un editor y la posibilidad de que el usuario proporcione nuevos atributos a la imagen exitosa.

**Precondición:** perfiles, texto de referencia de búsqueda.

**Postcondición:** resultados de la recuperación que puede ser o no imágenes exitosas y la actualización de atributos.

**Condición-terminación:** presentación de grupo de imágenes preseleccionadas y sus atributos con posibilidad de modificación o mensaje de no existencia de imágenes relacionadas.

**Tiempo-ejecución:** entre 10 y 60 segundos.

Las actividades como editar perfil, editar atributos de las imágenes no son tenidas en cuenta en este modelo debido a que sólo aquí se refiere a la comunicación posible entre agentes con su información preexistente, el modelo comunicación no incluye actividades que modifiquen las bases de datos.

La organización, como organización multiagente, esta expresada en las relaciones entre los agentes de los modelos de coordinación y comunicación; la organización

humana, como ya se había expuesto, permite a un experto en edición de materiales electrónicos o impresos y de concepto gráfico, proveerse de una herramienta de producción, quien sin ser necesariamente profesional en las áreas de trabajo, produce materiales de estas características y los expertos en contenidos no deben preocuparse por su edición y publicación, que generalmente es el gran problema y obstáculo en la implementación de materiales.

### 3.6 MODELO DE EXPERIENCIA

Según Glaser<sup>40</sup>, las metodologías de ingeniería del conocimiento pueden proveer una buena base para el modelado de sistemas multi-agentes, dado que ellos se ocupan del desarrollo de sistemas basados en el conocimiento, y como los agentes poseen características cognitivas, la metodología MAS-CommonKADS, puede proveer las técnicas para el modelado de estos agentes. La definición del conocimiento de un agente puede ser considerada como un proceso de adquisición de conocimiento.

La construcción de este modelo implica las siguientes actividades: descripción informal del dominio y tareas de la aplicación, que ya fue detallada en anteriores modelos; identificación de tareas genéricas, construcción del esquema del modelo, correspondencias entre papeles de tareas genéricas diferentes y el esquema del modelo.

Identificación de Tareas Genéricas y Esquema del modelo: La identificación de las tareas genéricas consiste en determinar cuáles de las tareas definidas en ese modelo requieren conocimiento:

AGENTE	TAREA GENÉRICA	CONOCIMIENTO
Usuario	<ul style="list-style-type: none"> <li>• Asignar perfiles.</li> <li>• Proporcionar criterios de búsqueda (texto)</li> <li>• Manejo de imágenes</li> <li>• Editar atributos y perfiles</li> </ul>	Ninguno
BDatos	<ul style="list-style-type: none"> <li>• Almacenar información</li> <li>• Manejo de la base de datos.</li> </ul>	Ninguno
Recuperación	<ul style="list-style-type: none"> <li>• Delimitación de imágenes a partir de perfiles de usuario y de trabajo</li> <li>• Procesamiento de texto.</li> <li>• Consulta y Actualización BDatos</li> </ul>	<ul style="list-style-type: none"> <li>• Personalización del aplicativo.</li> <li>• Aportar nuevos atributos de imagen.</li> <li>• Pesos de atributos, imágenes, categorías y perfiles de trabajo.</li> </ul>

<sup>40</sup> GLASSER, N. Op.cit.

El dominio, Recuperación de Información Gráfica, por sus especificidades, adquiere conocimiento a partir de dos fuentes: de una parte como producto de la utilización que se le de al sistema, como producto de los resultados exitosos que generan un cambio automático de la relevancia de los atributos frente a las imágenes, de las imágenes frente a sus categorías y de las categorías frente a los perfiles de trabajo, aplicando las formulas de cálculos de pesos y similitud del modelo vectorial. Como segundo es necesario considerar como conocimiento del sistema adquirido el que proviene de las decisiones tomadas por el usuario como experto en diseño de materiales y dotado de experiencia y concepto gráfico, que contribuye al enriquecimiento de los atributos de las imágenes.

Esta adquisición de conocimiento se consigue como consecuencia de la utilidad de la aplicación de consultas, para lo que se utilizan las siguientes operaciones:

Consultas basadas en retroalimentación del usuario. Esta estrategia selecciona términos importantes que han sido identificados por el usuario y que ayudan en una nueva formulación de la consulta. Esta operación aplica para resultados no exitosos de la recuperación y esta sujeta al concepto gráfico del usuario como fuente de conocimiento.

Consultas basadas en el análisis local de lo recuperado. A diferencia de la retroalimentación del usuario, esta técnica extiende el grupo de documentos relevantes en forma automática, al buscar términos que están relacionados a los criterios de búsqueda con la aplicación de lematización y tesauros.

Y por ultimo basados en el conjunto global de información existente en el banco de datos, referido en la cantidad y calidad de la información con la que se alimente la base de datos.

Todo lo anterior con el objeto de afinar y aumentar las operaciones exitosas y como consecuencia la calidad de los información asociada a cada criterio.

## **4. DISEÑO**

### **4.1 INTRODUCCIÓN**

Aplicar distintas técnicas y principios con el propósito de definir un producto con los suficientes detalles como para permitir su realización física, con el propósito de satisfacer determinada especificación de un sistema, que se ajuste a posibles limitaciones y que respete los requisitos sobre forma, rendimiento, utilización de recursos, costos, etc., son la base de un diseño exitoso.

El modelo de diseño para Mas-CommonKADS tiene como objetivo documentar todas las decisiones de diseño y determinar por una parte la arquitectura de la red de agentes, y por otra la arquitectura de agente más adecuada para cada agente. En el modelo de diseño deben considerarse los requisitos no funcionales del sistema y establecerse una relación entre los modelos del análisis y la arquitectura del agente, como paso previo a su implementación. Los requisitos no funcionales así como la arquitectura concreta que se definen pueden obligar a modificar partes de los modelos del análisis.

El diseño en MAS-CommonKADS incluye la descomposición del sistema en subsistemas, diseño de la interfaz gráfica de usuario y el diseño de la base de datos del sistema.

### **4.2 DISEÑO DE LOS AGENTES: DESCOMPOSICIÓN EN SUBSISTEMAS**

#### **4.2.1 Subsistema para la asignación de perfiles.**

##### 4.2.1.1 Perfil de usuario:

El modelado del perfil de usuario, como representación del usuario en las tareas computacionales, implica la adaptación en tres categorías: datos del usuario, que comprende sus características; datos de utilización, que comprende los datos de la interacción de los usuarios con el sistema, que no pueden ser consideradas características del usuario pero que pueden servir para la toma de decisiones de adaptación; y, los datos del ambiente que comprenden todos los aspectos del ambiente del usuario que no están directamente relacionados con él mismo.

El perfil del usuario va a contener características como las demográficas, que permitirán la personalización del aplicativo, creando una relación uno a uno para un número determinado de usuarios, dotando a cada uno de la información básica

de identificación y sus preferencias de usuario, a partir de su historial de trabajo, donde se registra los perfiles de trabajo utilizados, categorías y demás información que permita de una parte ubicar al usuario en las características de su última sesión de trabajo, y de otra sugerir perfiles de trabajo o datos de utilización.

Igualmente considera otras características como los objetivos y plan del usuario (tareas del usuario), más relacionada con el contexto de trabajo del usuario que con el usuario como individuo, que depende de por qué el usuario utiliza el sistema y qué pretende verdaderamente conseguir, los objetivos del usuario son la característica que puede cambiar de sesión de trabajo en sesión de trabajo, e incluso puede cambiar varias veces en la misma sesión. En esta propuesta se hace evidente por el hecho de que a pesar de que el usuario ya tenga determinada su área de trabajo y sus categorías, el documento ilustrado puede sufrir un cambio temático que implica la búsqueda de una imagen que no pertenece a los perfiles de trabajo predefinido inicialmente, característica que sería complementada con la experiencia e interés del usuario.

Para proporcionar los datos de usuario, la estrategia a seguir como método de adquisición es que el propio usuario provea los datos por ser la única fuente de información posible, de una parte los datos demográficos solicitados, y de otra, basados en su experiencia y conocimiento, con la creación de los perfiles de trabajo o datos de utilización, a partir de grupos de imágenes clasificadas por diferentes criterios y formatos.

Los datos de utilización pueden ser observados directamente y almacenados o adquiridos, que para este caso, es la clasificación en áreas de trabajo, clasificación que es creada por el usuario, de manera que puede crear tantas áreas de trabajo como combinaciones considere el usuario pueda crear para acercarse a sus necesidades de trabajo. Por ejemplo, la creación del área de trabajo Ciencias Naturales, ha de ser utilizada como herramienta de edición gráfica para la creación de materiales de esta área de formación, conformada por las categorías Plantas, Animales, etc., que a su vez pueden ser creadas tantas como imágenes se desee clasificar.

Para terminar, los datos del ambiente del usuario, que dotan al sistema de posibilidades de enriquecer la base de imágenes con nuevos y cada vez más preciso atributos de su descripción, contando con lo "aprendido" como producto de su utilización y por las interminables referencias de imágenes que cada día es más grande y que permite la creación de categorías cada vez más específicas.

#### 4.2.1.2 Perfil de trabajo:

Como ya se relacionó en el anterior perfil, sobre los datos de utilización del perfil de usuario, el perfil de trabajo es la personalización por grupos de categorías que sean requeridas para un área de trabajo determinada por el material a desarrollar. Un perfil de trabajo puede contar con las categorías que el usuario considere suficiente material para la elaboración de un material, además que una vez creado este perfil se tiene la posibilidad de adicionar o eliminar categorías haciendo flexible la conformación y definición de los perfiles de trabajo, y puede haber tantos perfiles de trabajo como diferentes áreas de trabajo pueda requerir un usuario.

Tanto el perfil de usuario como el de trabajo se basa en el modelo vectorial para calcular los pesos de los elementos que se le relacionan, que para el caso del perfil de usuario relaciona los perfiles de trabajo más usados o preferentes y el historial de uso "log" para evidenciar preferencias y recuperar sesiones de trabajo anteriores; y para el caso del perfil de trabajo, relaciona las categorías más utilizadas quienes a su vez relaciona las imágenes más utilizadas por categoría, consiguiendo entonces también imágenes más utilizadas.

### **4.2.2 Subsistema para el procesamiento de texto.**

#### 4.2.2.1 Análisis léxico del texto:

El preprocesado léxico inicial ha consistido en convertir la cadena de entrada, esto es, el texto de cada documento, en un conjunto de palabras o lemas<sup>41</sup>, que puedan servir luego como términos índice.

Es importante determinar qué palabras o lemas van a incluirse en el conjunto de lemas a tratar. Parece claro que cualquier palabra formada por caracteres alfabéticos debiera incluirse como candidato a término índice. Pero en la colección nos encontramos con otros caracteres (números, signos de puntuación, etc.). Relacionado con ello, un aspecto que debe tenerse muy en cuenta es el idioma de los documentos: la colección en español contiene caracteres como la ñe, vocales acentuadas, símbolos especiales, etc, que forma parte de los términos junto a caracteres ASCII.

Además un aspecto previo en esta fase consiste en identificar cada palabra del texto. Parece claro que el separador por excelencia es el espacio, pero también son separadores los signos de puntuación. En este apartado se trata cada caso:

---

<sup>41</sup> CHRISTOPHER, Fox. *Lexical Analysis and Stoplists*. En W.B. Frakes y R. Baeza-Yates. *Information retrieval, data structures and algorithms*. New Jersey, London, etc.: Prentice-Hall, 1992.

- Separación de palabras

Para separar los términos normalmente se utiliza el carácter espacio y los caracteres de puntuación.

Si se utiliza el espacio y los caracteres de puntuación, hay que reconocer cada uno de ellos como separadores y obrar en consecuencia. Por ejemplo, si los separadores están en el conjunto [.,,:;!¿?()/'"}] más el espacio, un texto como "¡Qué! No. Adiós, María." quedaría separado en las siguientes palabras "Qué No Adiós María". Ahora bien, muchas veces interesa conservar los separadores (son importantes para determinar dónde empieza y acaba una oración, por ejemplo), y quizás la salida del preprocesado sería conveniente que fuera "¡ Qué ! No . Adiós , María .". Después simplemente separamos por espacios y obtenemos todos los elementos (términos y signos de puntuación). El único inconveniente es especificar el conjunto de signos de puntuación, pero no hay mayor problema.

Este prototipo por su particularidad de requerir del párrafo sólo palabras índice a procesar, aplica una separación a partir de los caracteres de puntuación más el espacio, ya que el aplicativo no implica análisis sintáctico o semántico (no hay interpretación de textos desde el lenguaje natural), sin ser necesario diferenciar oraciones o párrafo, sólo toma un grupo de palabras a procesar, además es de tenerse en cuenta de que se predefine como idioma el Español.

- Tratamiento de acentos

El tratamiento de vocales acentuadas es también importante. La pregunta es: ¿incluimos acentos como aparecen, o los sustituimos por la vocal no acentuada? Por ejemplo, el buscador Google las separa, y no es lo mismo buscar por 'ángel' que por 'angel'. Otros sistemas de búsqueda menos potentes sustituyen las vocales acentuadas. En la política del analizador debe quedar claro qué se hace. Además en el preprocesado hay varias etapas como la eliminación de palabras vacías o la lematización, que pueden o no tener en cuenta las vocales acentuadas. Asimismo, es importante también determinar si en el almacenamiento de términos índice se utilizan o no las palabras con vocales acentuadas.

En general, las vocales acentuadas incluyen una carga semántica importante a la palabra, pero en recuperación de información no suelen considerarse; el motivo no es otro que el alto grado de errores ortográficos que se cometen con los acentos, por lo que, generalmente, en el preproceso léxico se convierten a vocales no acentuadas aquellas que lo estén en el texto.



En el prototipo aplica convertir vocales acentuadas y transformarlas a vocales sin acentos, para lo que los atributos de las imágenes son no acentuados para conseguir compatibilidad. Se ha optado siempre por convertir a formas no acentuadas.

- Internacionalización. Caracteres ISO-8859-1 (Latin-1)

Los caracteres del texto en español son los 128 caracteres del ASCII (ISO 646-1983 IRV) y los 128 segundos caracteres del ISO-8859-1 (ECMA-94 Right Part of Latin Alphabet Nr. 1). Con lo que se predetermina el español como idioma predeterminado.

- Tratamiento de números

Generalmente los números no se incluyen como términos índice, salvo que la colección de documentos sea legislativa (leyes, normas), de patentes, o en general, si la información numérica es de importancia. Sin embargo pudieran ser apropiados algunas veces cuando nos referimos a años, aunque para ello suele ser habitual incluir un campo explícito en la base de datos.

Si se decide indexar los números, el problema es que el volumen de términos índice se dispara (por ejemplo, si se incluye cualquier número de 6 cifras, potencialmente el número de términos se incrementa en 1 millón).

El prototipo especifica no incluir los números como términos índice. En este caso, se deben eliminar los números y aquellos que puedan contener puntos, comas o comillas (1.000.000, 123,45, 3'75).

Para el caso de las cadenas alfanuméricas (por ejemplo, en sub-21, EEUU'94, B12, PS/2, SA-0992-G, 27-E, 180-R), se aplicará un procedimiento de eliminación de estas cadenas, ya que las expresiones alfanuméricas no es información relevante ante los criterios de búsqueda de las imágenes.

- Detección de nombres propios

Si son importantes para la recuperación de información los términos que actúan gramaticalmente como nombres, más importantes son las palabras que actúan como nombres propios. Muchas veces todo un documento puede representarse por los nombres propios que contiene, y a menudo se les asigna mayor peso.

Sencillamente este prototipo no requiere que sean considerados los nombres propios como relevantes en la determinación de criterios o atributos de selección

ya que los atributos de las imágenes están conformados por palabras genéricas que generalmente son características de ella.

- Detección de siglas

Las siglas pueden considerarse un caso particular de nombre propio, en el que todos los caracteres aparecen en mayúscula. En el preprocesado léxico se incluye una subrutina que marca las siglas en la cadena de texto pasada como argumento.

Se considera que una sigla está formada por las iniciales de las palabras que representa, separadas por punto, o todas juntas sin punto (excepto cuando se trate de un único carácter en mayúsculas, que no se considera sigla si no lleva punto).

El prototipo simplifica esta rutina, al igual que para palabras vacías (que se especifica más adelante), con una tabla con todas las siglas conocidas, unidas y separadas con puntos, y su eliminación de las palabras seleccionadas, por no contener relevancia ante los atributos de la imagen. Esta rutina se debe aplicar antes de convertir en minúsculas el grupo de palabras seleccionadas.

#### 4.2.2.2 Eliminación de palabras vacías

Con el objetivo de reducir el número de términos índice no se incluyen palabras que, por su poca capacidad semántica o por su alta frecuencia, son poco significativas en el proceso de recuperación de información. Este conjunto de palabras se compone de preposiciones, artículos, adverbios, conjunciones, posesivos, demostrativos, pronombres, verbos (ser, estar) y algunos nombres, y se denomina conjunto de palabras vacías (stop words en inglés)

Con la eliminación de palabras vacías se pretende reducir el ruido que pudieran introducir en la recuperación. Es una forma de delimitar el número de términos que servirán como términos índice. La mayoría de motores de búsqueda en Internet eliminan las palabras vacías de contenido.

Para la eliminación de palabras vacías se ha separado el texto en palabras y luego se han buscado en la lista de palabras vacías, utilizando para ello el algoritmo de comparación.

#### 4.2.2.3 Lematización

Se denomina lematización al proceso mediante el cual se buscan variaciones morfológicas de los términos, con el objetivo de extraer la raíz común a ellos. La

forma canónica de la raíz se denomina lema (aunque el término no está aceptado por la Real Academia de la Lengua), y representa las variaciones de los términos que de ella derivan. Generalmente la lematización se realiza porque semánticamente los términos que derivan de una raíz están muy próximos entre sí.

Para la lematización se utiliza un lematizador. El lematizador detecta variantes tanto flexivas como derivativas de una palabra. Ello implica algoritmos más o menos complejos, desde los que juegan con las variantes de género y número para la lematización flexiva, a los que combinan éstos con una serie de sufijos y prefijos, para la lematización derivativa.

El objetivo de la lematización en recuperación de información es reducir el número de términos índice semánticamente muy parecidos. Con ello se pretende aumentar las tasas de exhaustividad del sistema de recuperación, y a la vez reducir el espacio de almacenamiento y aumentar la velocidad del proceso.

Es obvio que las palabras que pertenecen a una misma familia están relacionadas semánticamente, pero estas relaciones no se ponen de manifiesto de manera automática en los sistemas convencionales de RI. Cuando se utiliza el modelo vectorial como algoritmo del sistema de recuperación se trabaja con la frecuencia de aparición de términos en el documento. Los términos se tratan como meros conjuntos de caracteres, y por tanto, un término en singular es distinto al término en plural, lo cual puede provocar perder documentos relevantes. Con ello los métodos basados en frecuencias arrastran un error inherente a su propia construcción. Con la lematización se pretende reducir ese error, de modo que varias palabras con un origen común se puedan reducir a un único término, basándose en que todas guardan relación semántica.

La forma de implementar el proceso lematizador en el sistema es el siguiente. Se obtienen las palabras del documento, y se pasan al lematizador flexivo y derivativo, obteniendo una tabla en cada caso con la palabra origen y lematizada. A la hora de aplicar la lematización en el sistema, para cada palabra se obtiene su lematizada y se sigue el procesamiento léxico.

Un tema que parece claro es que los términos marcados como nombres propios o siglas no deben ser lematizados en el proceso, las cuales para este prototipo son eliminadas previamente. Un apunte más en relación con la eliminación de palabras vacías: hay que aplicar el algoritmo de eliminar antes de la lematización, y después eliminar palabras repetidas consecuencia de la lematización.

El prototipo se apoya en un tipo de diccionario con palabras genéricas y un grupo de palabras consideradas derivadas o relacionadas, palabras entre las que se

determina los criterios relevantes de búsqueda. Es decir si una o varias palabras se encuentran entre las derivadas, se convierte en la palabra genérica, obteniendo para el caso de que sean varias, palabras repetidas a eliminar. Es de tener en cuenta que los atributos de las imágenes esta conformado por sólo palabras genéricas. Similar al procesamiento tesauro.

#### 4.2.2.4 Tesauro

Un tesauro es un diccionario de términos controlados con relaciones explícitas entre los términos. Por ejemplo, un tesauro geográfico contiene nombres de poblaciones y accidentes geográficos, así como las relaciones entre ellos (por ejemplo, Salamanca estará relacionado con España). Otros tesauros están contruidos como diccionarios de sinónimos y/o antónimos.

Normalmente los tesauros se utilizan en recuperación de información para ampliar las consultas, de modo que, además de buscar por los términos que aparecen en la misma, se buscan por aquellos términos relacionados en el tesauro. Normalmente esto produce mejores valores de exhaustividad.

En tesauros, de forma similar que para lematizaciones, se crea un diccionario que en este caso es de palabras sinónimas a una genérica, no derivadas o relacionadas como es el caso de lematizaciones, que igualmente transforma las palabras seleccionadas en genéricas y que necesariamente también condiciona el hecho de que los atributos de las imágenes son palabras genéricas.

#### 4.2.2.5 Eliminación de palabras repetidas

Una vez calculado el peso de las palabras seleccionadas, donde tiene como factor su frecuencia, se procede a eliminar. Es de tener en cuenta que para que esto sea posible se debe aplicar primero los procesos de lematización y tesauro, se calcula el peso de la palabra y se eliminan.

### **4.2.3 Subsistema para la administración de la base de datos y RIG.**

Anterior al procesado léxico del texto y a las consultas es necesario almacenar la información en bases de datos, indexarlos para aumentar la velocidad e implementar un algoritmo que compare el listado de palabras con la base de datos para extraer los relacionados.

El proceso de almacenamiento, búsqueda de imágenes, y de registro de pesos se realiza por medio de una base de datos la cual está orientada bajo el modelo entidad relación en el cual se estructura la información que maneja el sistema,

donde cada registro hace las veces de arreglo para la aplicación del modelo vectorial.

#### 4.2.3.1 Administración de la base de datos.

La base de datos almacena la información relacionada con las imágenes, como atributos, categorías, perfiles y las rutas de ubicación de las imágenes que está organizadas en carpetas por categoría que serán ejecutados desde la base de datos.

La administración de la base de datos permite aplicar sus tareas básicas como creación, modificación o eliminación de la información allí contenida para efectos de actualizarla, complementarla y permitir su crecimiento y depuración, así como enriquecer los diccionarios para lematizaciones y tesauros.

#### 4.2.3.2 Generación de pesos

La relevancia de los atributos de imagen, las imágenes, las categorías y perfiles más utilizados son pesados según coincidencias producto del éxito de cada recuperación aplicando el modelo vectorial, y almacenados en los campos según sea el factor calculado.

#### 4.2.3.3 Comparación

El prototipo toma como base un arreglo de entrada producto de la división de palabras del párrafo o texto objeto de búsqueda, a este arreglo se le aplica el procesamiento de texto, y sólo cuando se obtiene el conjunto de palabras consideradas relevantes en un nuevo arreglo, se compara por medio de consultas con los atributos relacionados a cada imagen.

#### 4.2.3.4 Actualización de la base

El sistema actualiza automáticamente el orden de los atributos por el peso que se le asigne por haber coincidido con el criterio de búsqueda y su repetición, permite que las imágenes automáticamente se ordenen por pesos también por la suma de los pesos de sus atributos coincidentes, y a la vez que las categorías se ordenen por pesos como consecuencia de los pesos de las imágenes que la conforman y fueron exitosas, y por último la ordenación de los perfiles de trabajo por los pesos que adquieren por la suma de los pesos de las categorías que los conforman.

Adicionalmente permite la actualización, agregar, eliminar y modificar, de los atributos (palabras genéricas) por el usuario en el sentido de que él conocerá los

resultados de las búsquedas, los criterios de búsqueda (palabras relevantes del texto) y las palabras que fueron relevantes pero que no tuvieron coincidencias con los atributos de la imagen y que pueden ser representantes de la imagen exitosa, para su ingreso; o evidenciar la no pertinencia de algunos de los atributos predefinidos de una imagen para eliminarlo, o por algún motivo requiera que uno de ellos sea modificado. No se permitió el ingreso automático a los atributos a la base de datos de las palabras genéricas no coincidentes por cuanto se permitiría ingreso de una gran cantidad de atributos no controlados y seguramente no relacionados con las imágenes que sólo van a producir ruidos o basuras de procesamiento sin aportar al sistema.

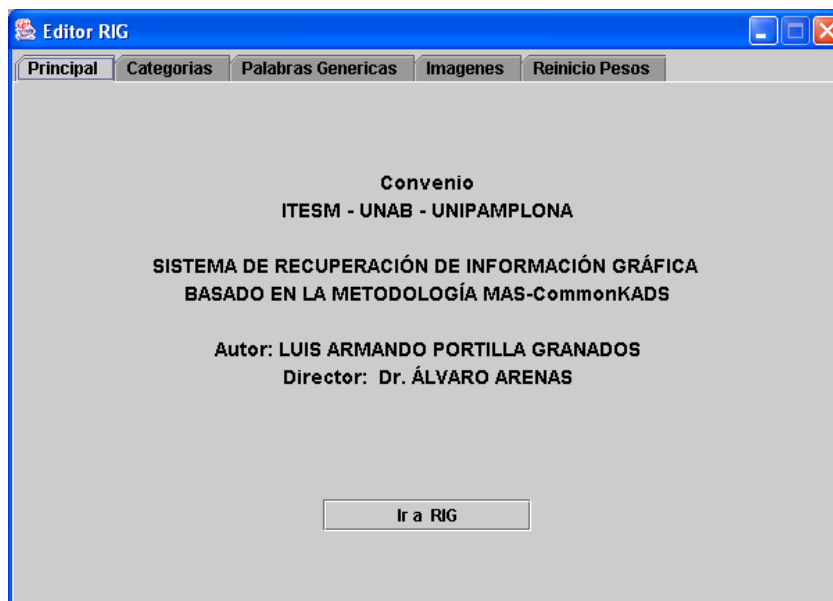
### 4.3 DISEÑO DE LA INTERFAZ GRÁFICA DE USUARIO

Por fines didácticos y demostrativos de la ejecución del prototipo, se visualizará los eventos que se ejecutan desde el texto base, el resultado del procesamiento del texto y la entrega de resultados, además del módulo de administración de la base de datos.

Se utiliza dos módulos, uno para la Recuperación de Información Gráfica (RIG) y otro para la Administración de la Base de Datos (Edit RIG). Los módulos se distribuyen por pestañas en las que se disponen tareas específicas y los pasos de la ejecución del aplicativo.

#### 4.3.1 Administración de la BDatos (Editor RIG).

Figura 11. Ventana editor RIG.

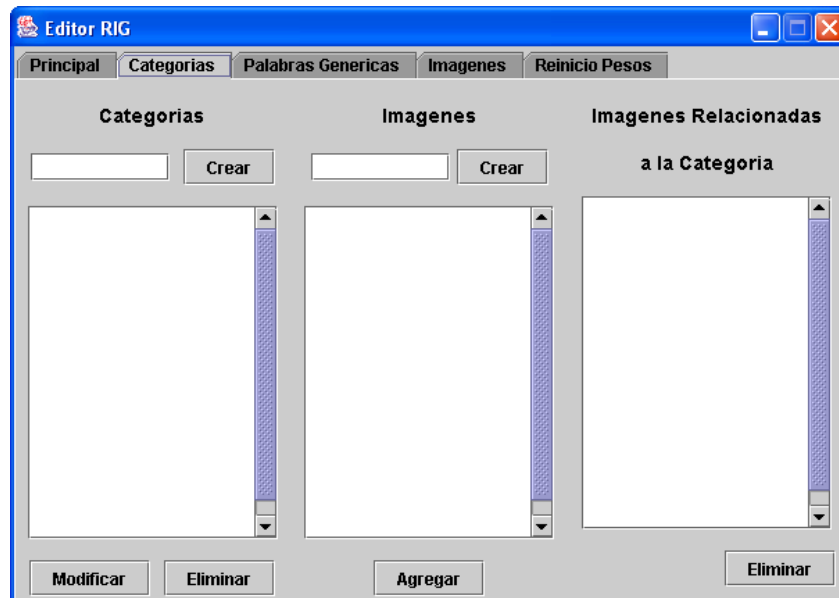


Este módulo permite la administración de la base de datos aplicando las funciones básicas de creación, modificación y eliminación de la información contenida. Se distribuye en las secciones de categorías, palabras, imágenes, y reinicio de pesos, que serán detalladas en los siguientes apartes:

### **Ventana para Edición de Categorías:**

Esta ventana está distribuida en tres columnas, la primera muestra la lista de las categorías existente y las opciones *Crear* para agregar nuevas categorías, *Modificar* que permite cambiar el nombre de una categoría, y *Eliminar* que suprime una categoría, la segunda columna permite crear, eliminar o agregar imágenes, y la tercera, visualiza la lista de imágenes relacionadas a la categoría seleccionada.

Figura 12. Ventana para edición de categorías.



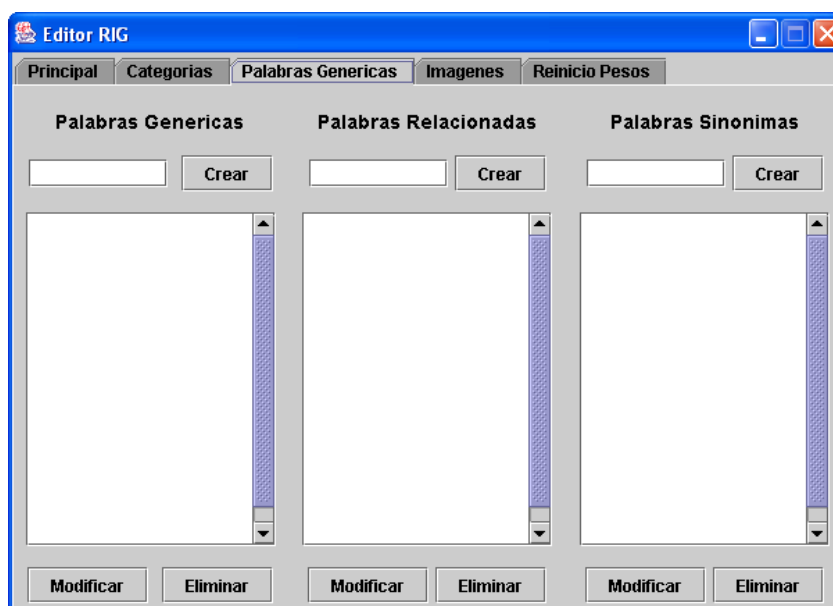
Esta sección permite realizar varios procedimientos así:

- Crear una categoría y asociarle imágenes.
- Seleccionar una categoría existente para agregar o quitar imágenes.
- Crear y eliminar imágenes para lo que sólo asigna el nombre de la imagen y en un paso posterior en la sección *Imágenes* le asigna la ruta del archivo de la imagen y los atributos.
- Modificar el grupo de imágenes asociadas a una categoría por eliminación de imágenes.

### **Ventana Edición de Palabras Genéricas, Derivadas y Sinónimas:**

Esta sección ofrece las operaciones de edición de palabras genéricas o atributos, presentada en tres columnas, la primera con el listado de palabras genéricas existentes, la segunda presenta las palabras relacionadas o derivadas (lematización), y la tercera las sinónimas (tesauros) asociadas a cada una de las genéricas.

Figura 13. Ventana edición de palabras genéricas, derivadas y sinónimas.



Que permite los siguientes procesos:

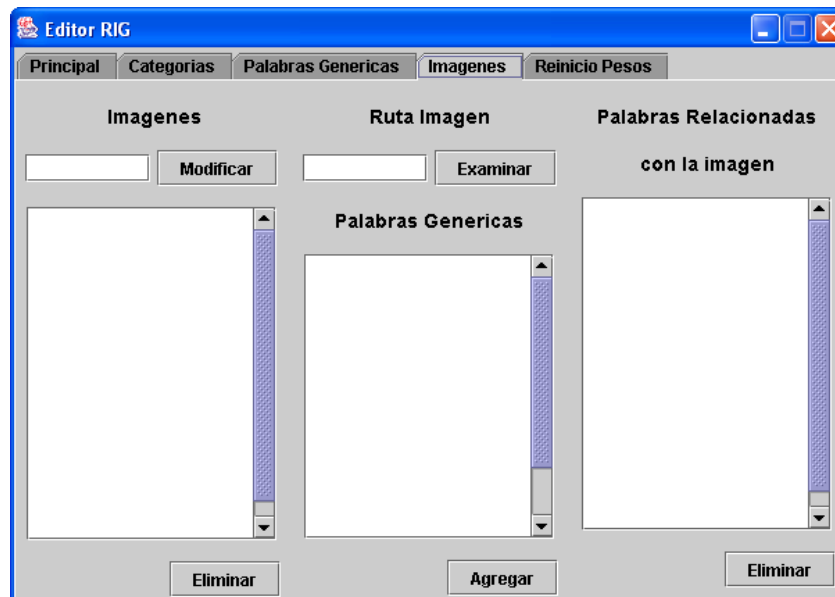
- Crear, modificar o eliminar palabra genérica.
- Crear, modificar o eliminar palabras relacionadas.
- Crear, modificar o eliminar palabras sinónimas.

### **Ventana para Edición de Información de Imágenes:**

En la sección de imágenes permite editar la información sobre imágenes. Se distribuye igualmente en tres columnas, la primera para las imágenes existentes; en la segunda columna se presenta las rutas de acceso a las imágenes, y el listado de palabras genéricas; y la tercera columna va mostrando el listado de atributos con que cuenta cada imagen.



Figura 14. Ventana para edición de información de imágenes.



- Modificar el nombre de una imagen, o eliminarla.
- Crear ruta de acceso con los archivos de las imágenes.
- Asignar y eliminar atributos a las imágenes.

### Ventana Demostrativa de Pesos:

Figura 15. Ventana demostrativa de pesos.

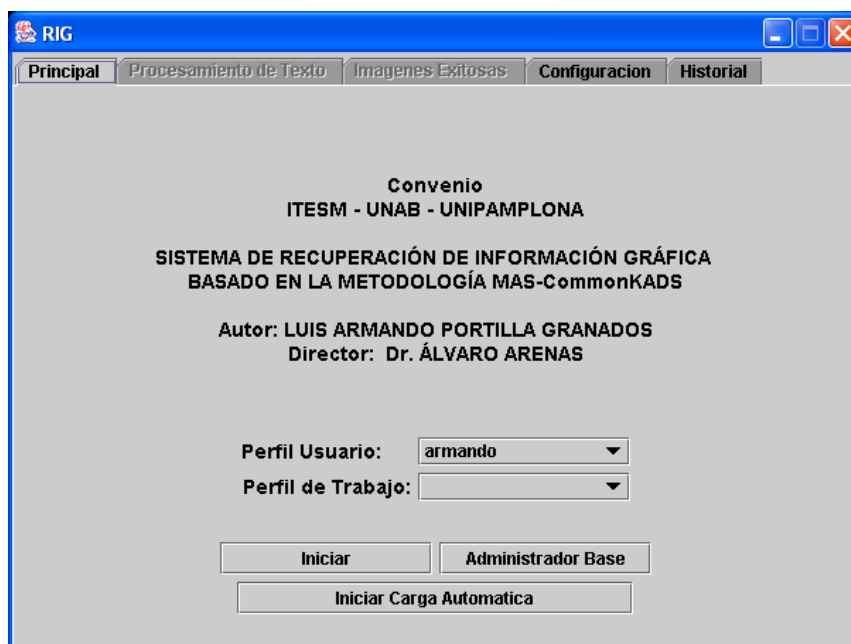


El usuario desde su perfil de usuario, al ingresar a esta sección puede ver los pesos de sus perfiles de trabajo, de las categorías asociadas a cada perfil de trabajo y de las imágenes relacionadas a cada categoría. Esta sección permite reinicializar los pesos de cualquiera de los listados para fines experimentales de comprobar los cambios que se generan sobre cada caso según los pesos que se adquieren como resultado de la utilización del prototipo.

### 4.3.2 Recuperación de Información Gráfica (RIG)

Este módulo ofrece las opciones para la Recuperación de Información Gráfica (RIG) en la que el usuario carga los perfiles de una sesión de trabajo y ejecuta las operaciones de procesamiento de texto, presentación de imágenes exitosas, configuración de perfiles e historial de las sesiones de trabajo realizadas.

Figura 16. Ventana Recuperación de Información Gráfica – RIG.



En la pestaña Principal se hace la presentación del prototipo y permite las siguientes funciones:

- Selección de los perfiles de usuario y de trabajo: para cada ejecución el sistema propone los últimos perfiles utilizados según el peso que haya adquirido como producto de su utilización.
- Iniciar: una vez seleccionados los perfiles, desde este botón inicia la ejecución del programa paso a paso y trae como resultado imágenes preseleccionadas.

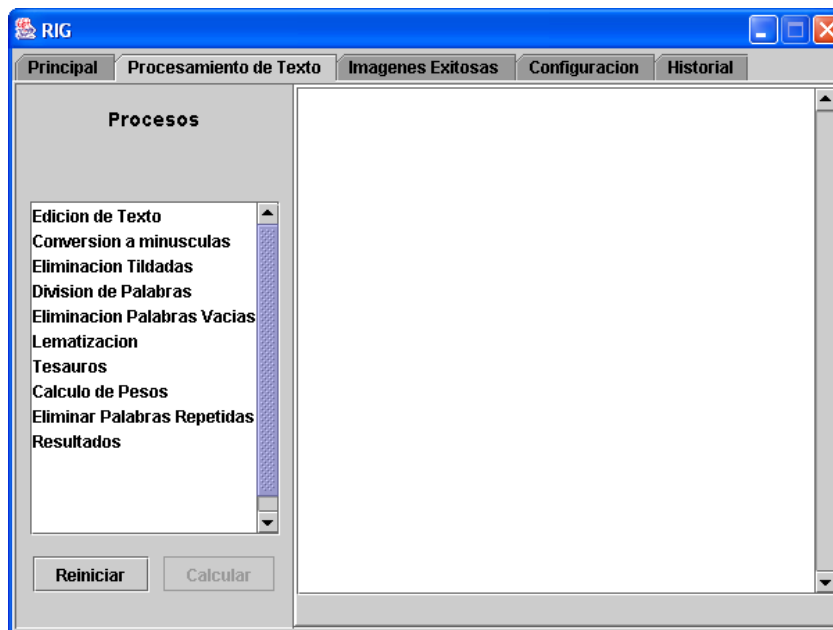
- Administrado de la base de datos: punto de enlace entre este módulo y el módulo de administración de la base de datos.
- Iniciar en Carga Automática de Recuperación: activando esta operación carga la imagen exitosa (de mayor peso) directamente desde el programa de trabajo cargando el texto en el portapapeles (Control C).

### Ventana para Procesamiento de Texto:

Esta sección ejecuta paso a paso los procesos necesarios para el procesamiento de texto que se distribuye en dos columnas, a la izquierda muestra el proceso ejecutado y a la derecha los resultados. Los procesos se listan según el orden de ejecución y permite visualizar los resultados de cada uno seleccionándolo.

La columna de la izquierda contiene dos operaciones: *Reiniciar* que posibilita reejecutar todos los procesos del procesamiento de texto, donde se considera la opción de ingresar nuevos parámetros de búsqueda, y *Calcular* que iniciar la ejecución de los procesos.

Figura 17. Ventana para procesamiento de texto.



El procesamiento de texto ejecuta los siguientes procesos:

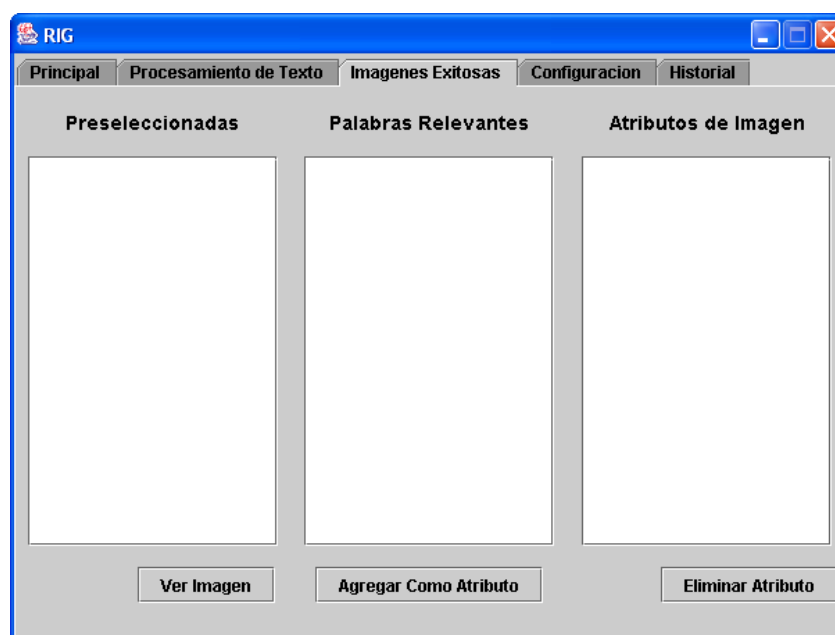
- Captura del texto base de procesamiento
- Conversión a minúsculas
- Conversión de caracteres especiales (tildadas y símbolos especiales)

- División de palabras
- Eliminación de palabras vacías, siglas, acrónimos y abreviaturas
- Aplicación de lematización
- Aplicación de tesauros
- Asignación de pesos a palabras relevantes
- Eliminación de palabras repetidas
- Presentación de los resultados del procesamiento de texto

### **Ventana de Presentación de Imágenes Exitosas:**

La presentación de imágenes preseleccionadas como producto del procesamiento de texto se hace en una nueva sección para posibilitar la visualización del listado de los nombre de las imágenes ordenado descendientemente según sus pesos, el listado de palabras relevantes (producto del procesamiento del texto), y listado de los atributos asociados a la imagen seleccionada, en tres columnas.

Figura 18. Ventana de presentación de imágenes exitosas.



Que permite los siguientes procesos:

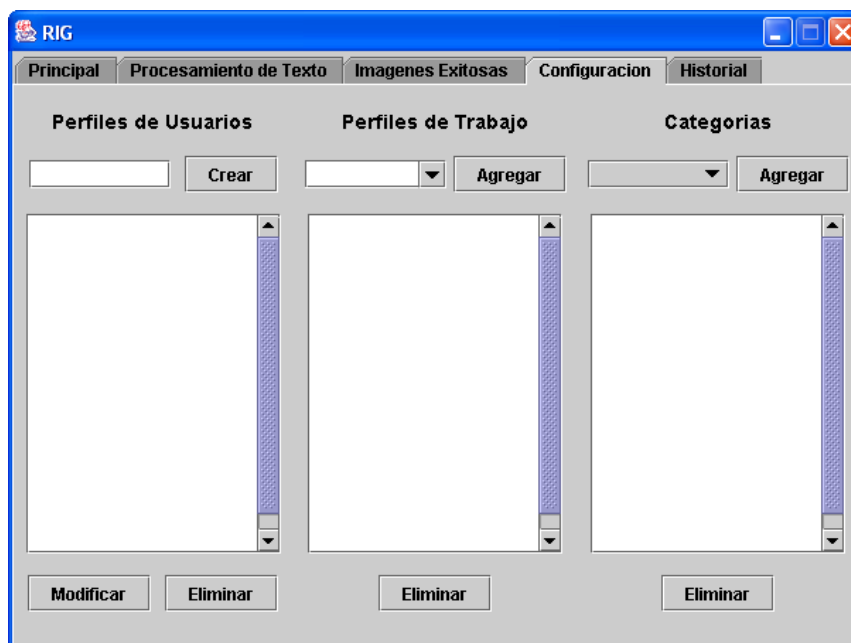
- Visualizar el listado de los nombres de las imágenes preseleccionadas.
- Visualizar la imagen seleccionándola y utilizando el botón *Ver Imagen*. Esta operación despliega otra ventana con la imagen que seleccione desde donde se puede copiar la imagen exitosa en el editor que se encuentre trabajando según el criterio del usuario.

- Visualizar el listado de atributos relacionados a la imagen seleccionada con la posibilidad de eliminar aquellos que el usuario considere no relevantes.
- Editar los atributos de la imagen seleccionada agregando de listado de palabras relevantes, producto del procesamiento de texto.

### Ventana para la Configuración de Perfiles:

El usuario tiene la posibilidad inicial de personalizar el aplicativo seleccionando los perfiles de usuario y de trabajo. Si la sesión de trabajo a iniciar no cuenta con los perfiles requeridos, el aplicativo ofrece en esta sección las opciones de configuración de perfiles, distribuidas en tres columnas, la primera para lo relacionado a perfiles de usuario, la segunda perfiles de trabajo y la tercera que visualiza las categorías existentes, que de ser necesario pueden ser editadas en el módulo de administración de la base de datos.

Figura 19. Ventana para la configuración de perfiles.



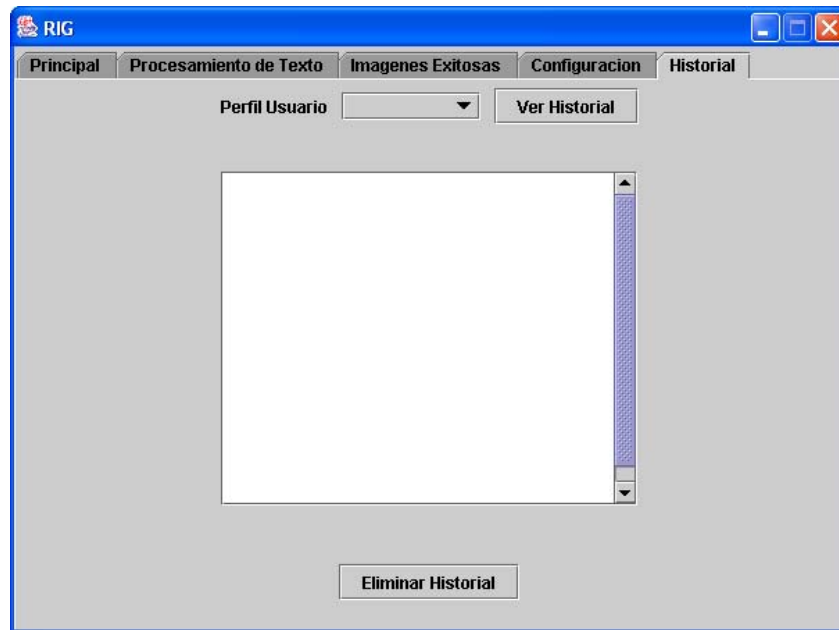
Esta sección ofrece las siguientes opciones:

- Crear, modificar o eliminar el nombre de los usuarios.
- Crear, agregar o eliminar un perfil de trabajo de los existentes al usuario seleccionado.
- Agregar o eliminar categorías al perfil de trabajo seleccionado.

### **Ventana de Historial de Usuario:**

Esta ventana presenta las sesiones de trabajo del perfil de usuario que seleccione. Esta información presenta la o las fecha de sesión y el o los perfiles de trabajo utilizados.

Figura 20. Ventana de historial de usuario.



Como proceso permite:

- Visualizar historial por usuario.
- Borrar un historial utilizando el botón *Eliminar Historial*.

## **4.4 DISEÑO DE LA BASE DE DATOS DEL SISTEMA**

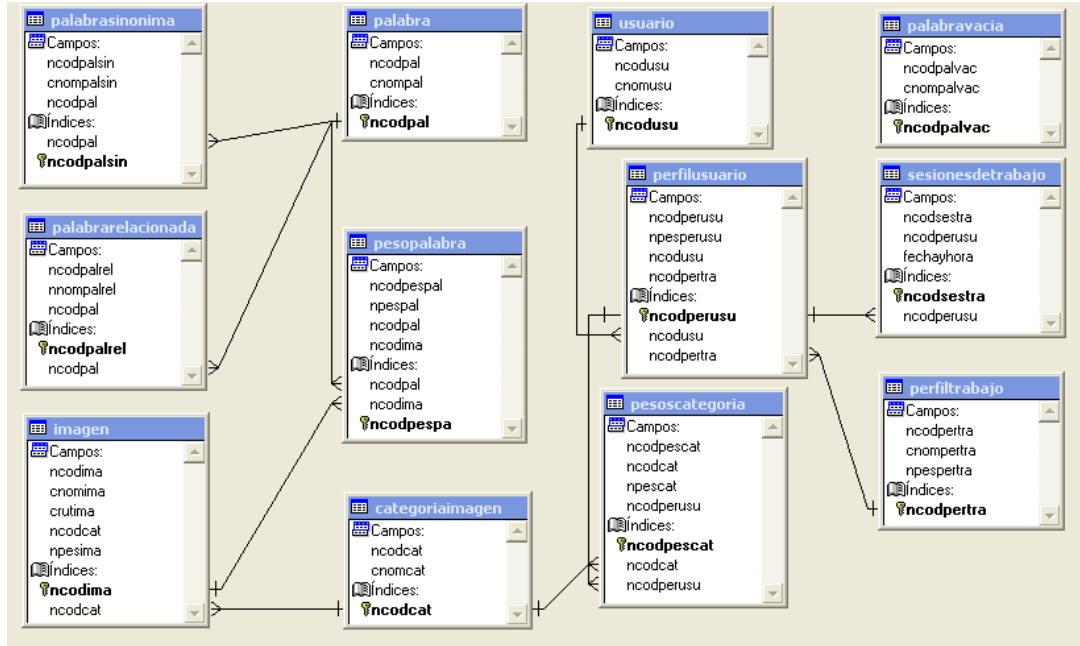
### **4.4.1 Modelo.**

Con esta técnica de diseño de bases de datos gráfica, que incorpora información relativa a los datos y la relación existente entre ellos, para poder así plasmar una visión del prototipo. Sus características fundamentales son:

- Reflejan tan sólo la existencia de los datos sin expresar lo que se hace con ellos.
- Es independiente de las bases de datos y de los sistemas operativos.

- Incluye todos los datos que se estudian sin tener en cuenta las aplicaciones que se van a tratar.

Figura 21. Modelo entidad relación.



#### 4.4.2 Descripción del modelo.

##### Tabla *Imagen*

Descripción: almacena la identificación por código de la imagen, la categoría a la que pertenece, su peso y la ruta de ubicación de archivo de la imagen. Cada imagen pertenece a una única categoría.

CAMPO	TIPO	DESCRIPCIÓN
ncodima	Numérico	Código de la imagen
cnomima	Carácter	Nombre de la imagen
crutima	Carácter	Ruta donde aloja la imagen
ncodcat	Numérico	Código de la categoría a la que pertenece
npesima	Numérico	Peso de la imagen

### **Tabla PerfilUsuario**

Descripción: información relacionada con el usuario, perfiles de trabajo y peso asociado a él según su uso. Permite que una categoría pertenezca a varios perfiles de trabajo y que cada usuario tenga varios perfiles de trabajo.

CAMPO	TIPO	DESCRIPCIÓN
ncodperusu	Numérico	Código del perfil usuario
npesperusu	Numérico	Peso de perfil de usuario
ncodusu	Numérico	Código de usuario
ncodpertra	Numérico	Código de perfil de trabajo

### **Tabla Palabra**

Descripción: almacena la palabras característica o atributo que tiene cada imagen. Estas palabras son genéricas y corresponden al tipo de palabras procesadas para su comparación, están en minúscula y no están tildadas. Las palabras están relacionadas a sus imágenes y pueden pertenecer a diferentes imágenes.

CAMPO	TIPO	DESCRIPCIÓN
ncodpal	Numérico	Código de la palabra
cnompal	Carácter	Palabra

### **Tabla PesoPalabra**

Descripción: contiene los pesos de las palabras según la imagen a la que esté asociada. Estos pesos se acumulan según su coincidencia con las palabras procesadas cuando la imagen resulta exitosa. Una palabra puede estar en distintas imágenes y con peso particular para cada caso.

CAMPO	TIPO	DESCRIPCIÓN
ncodpespal	Numérico	Código del peso la palabra
npespal	Carácter	Peso de la palabra
ncodpal	Numérico	Código de la palabra
ncodima	Numérico	Código de imagen

### **Tabla CategoriaImagen**

Descripción: contiene las diferentes categorías en las que se considere se puedan clasificar las imágenes. Una categoría puede hacer parte de varios perfiles de trabajo.



CAMPO	TIPO	DESCRIPCIÓN
ncodcat	Numérico	Código de la categoría
cnomcat	Carácter	Nombre de la categoría

### **Tabla *Usuario***

Descripción: registra los usuarios del aplicativo. Cada usuario puede determinar sus perfiles de trabajo.

CAMPO	TIPO	DESCRIPCIÓN
ncodusu	Numérico	Código usuario
cnomusu	Carácter	Nombre usuario

### **Tabla *PalabraSinónima***

Descripción: contiene las palabras relacionadas a las palabras genéricas o atributos de la imagen como base para aplicar tesauros (sinónimos). Cada palabra genérica tiene asociadas palabras sinónimas únicas, es decir las palabras sinónimas están asociadas con una sola palabra genérica.

CAMPO	TIPO	DESCRIPCIÓN
ncodpalsin	Numérico	Código palabra sinónima
cnompalsin	Carácter	Palabra sinónima
ncodpal	Numérico	Código de la palabra genérica

### **Tabla *PalabraRelacionada***

Descripción: contiene las palabras relacionadas a las palabras genéricas o atributos de la imagen como base para aplicar lematización (palabras relacionadas). Cada palabra genérica tiene asociadas palabras relacionadas únicas, es decir las palabras relacionadas están asociadas con una sola palabra genérica.

CAMPO	TIPO	DESCRIPCIÓN
ncodpalrel	Numérico	Código palabra relacionada
cnompalrel	Carácter	Palabra relacionada
ncodpal	Numérico	Código de la palabra genérica

### **Tabla *PesosCategoria***

Descripción: las categorías adquieren un peso por su uso según resulten exitosas las imágenes asociadas a cada una. Una categoría puede adquirir varios pesos según perfil de trabajo y usuario a que pertenezca.

CAMPO	TIPO	DESCRIPCIÓN
ncodpescat	Numérico	Código peso de categoría
ncodcat	Numérico	Código de la categoría
npescat	Numérico	Peso de la categoría
ncodperusu	Numérico	Código del perfil de usuario

### **Tabla *PerfilTrabajo***

Descripción: incluye los perfiles de trabajo creados como conjunto de categorías de un área determinada. Varios perfiles de trabajo pueden tener en común categorías y un usuario puede tener varios perfiles de trabajo.

CAMPO	TIPO	DESCRIPCIÓN
ncodpertra	Numérico	Código de perfil de trabajo
cnompertra	Carácter	Nombre del perfil de trabajo
npespertra	Numérico	Peso del perfil de trabajo

### **Tabla *SesionesDeTrabajo***

Descripción: guarda el historial de las sesiones de trabajo y permite proponer la última sesión cada vez que cargue el aplicativo. Cada usuario puede registrar varias sesiones con diferentes perfiles de trabajo.

CAMPO	TIPO	DESCRIPCIÓN
ncodsestra	Numérico	Código de la sesión de trabajo
ncodperusu	Numérico	Código del perfil de usuario
fechayhora	Fecha y hora	Fecha y hora de la sesión de trabajo

### **Tabla *PalabraVacía***

Descripción: contiene las palabras que semánticamente se consideran vacías, incluye para este caso palabras alfanuméricas, siglas, acrónimos y abreviaturas.

CAMPO	TIPO	DESCRIPCIÓN
ncodpalvac	Numérico	Código de palabra vacía
cnompalvac	Carácter	Palabra vacía

## **4.5 DISEÑO DE LA PLATAFORMA**

### **Descripción**

No se utiliza plataforma multiagente predefinida.

### **Usa-lenguaje**

Java

### **Hardware-requerido**

PC Pentium III, RAM 128MB, DD 20GB, o superior.

### **Software-requerido**

SO Windows, Visual Fox Pro 6.0, JCreator Pro.

### **Usuario**

Especificador

## **5. DESARROLLO**

### **5.1 INTRODUCCIÓN**

El desarrollo describe los procesos de análisis, diseño y programación del prototipo, utilizando herramientas, técnicas y metodologías necesarias

Durante la etapa del Desarrollo del Sistema de Información se materializa el prototipo de acuerdo a lo establecido en el Diseño Conceptual del Sistema. También durante esta fase se aplicarán en detalle algunos de los procesos y cálculos que se definieron a nivel conceptual durante las etapas de análisis y diseño o deberán tenerse definidos aquellos que surgieron de nuevas políticas establecidas.

También en esta fase se deberán de integrar la operatividad de las funciones y pruebas con situaciones reales de aplicación con evaluaciones periódicas de resultados exitosos y no exitosos. Igualmente se explica el método iterativo y las responsabilidades a los usuarios ya que el usuario participa directamente en todo el proceso.

En el desarrollo de un prototipo se preparan los siguientes componentes:

- El lenguaje para el diálogo o conversación entre el usuario y el sistema.
- Pantallas y formatos para la entrada de datos.
- Módulos esenciales de procesamientos.
- Salidas del sistema.

La incorporación en la interfaz de entrada/salida de características representativas de las que serán incluidas en el sistema final permite una mayor exactitud en el proceso de evaluación.

Para tal fin se describen, fundamentalmente, métodos de análisis y diseño estructurado, sus herramientas (gráficas y textuales) de modelamiento y especificación, y criterios de aplicación de las mismas. El método esta basado en la construcción de un conjunto de modelos, a partir del modelo esencial, tendientes a describir, especificar y documentar todas las decisiones de diseño

tomadas durante la búsqueda de una solución al problema especificado mediante dicho modelo esencial.

La Ingeniería de Software Orientada a Agentes se soporta de las metodologías orientadas a objetos<sup>42</sup>, de lo que se pueden citar varias razones que justifican la extensión de las metodologías orientadas a objetos.

- Existen similitudes entre el paradigma orientado a objetos y el paradigma orientado a agentes. Como establece Shoham<sup>43</sup>, los agentes pueden ser considerados objetos activos, objetos con estados mentales.
- El uso común de los lenguajes orientados a objetos para la implementación de los agentes.
- La popularidad de las metodologías orientadas a objetos.

De lo anterior el presente trabajo basó la aplicación de la metodología Orientada a Agentes MAS-CommonKADS para el análisis y diseño. El desarrollo trabaja con programación orientada a objetos.

Para el desarrollo del prototipo se usa el lenguaje de programación Java, que es utilizado en Internet y que actualmente es uno de los lenguajes más poderosos y populares del mundo. El entorno de edición de Java es Jcreator que incluye diversas funciones como un gestor de proyectos, un entorno configurable por el usuario de subrayado de sintaxis, un visor de clases, un compilador múltiple, líneas numeradas, y muchas otras funciones más. La versión Jcreator 3.0 Pro incluye adicionalmente un entorno configurable por el usuario de subrayado de sintaxis.

Para el diseño de la base de datos se utiliza Visual FoxPro 6.0, que es un sistema poderoso de desarrollo de bases de datos para crear rápidamente aplicaciones de bases de datos de alto desempeño para equipos de escritorio de cliente enriquecido, cliente distribuido, cliente/servidor y aplicaciones Web,

A continuación se describe las clases que utiliza la estructura correspondiente a los agentes *Bdatos* y *Recuperar* diseñados en los anteriores capítulos, y sus tareas representadas en las funciones que las conforman. El código se adjunta en los archivos *EditRIG* y *RIG* que corre en JCreator Pro.

---

<sup>42</sup> GROSSE, A. Op.cit.

<sup>43</sup> SHOHAM, Y., "Agent-oriented programming". Artificial Intelligence, Vol. 60, Nº. 1, pp. 51-92, March 1993.

## 5.2 DECLARACIÓN DE LAS CLASES

### El encabezado: librerías o paquetes de clases utilizados:

```
import java.lang.*;
    Proporciona funcionalidad básica. (Está contenida en Java)
import java.io.*;
    Datos de entrada y salida.
import java.util.*;
    Contiene utilidades de uso común.
import java.awt.*;
    Proporciona capacidades para interfaces gráficas de usuario.
import java.awt.font.*;
    Representa varios tipos de letra.
import java.awt.event.*;
    Gestiona eventos.
import java.awt.dnd.*;
    Soporta operaciones de arrastrar – y - dejar.
import java.sql.*;
    Se comunica con una base de datos SQL (Structured Query Language)
import java.beans.*;
    Permite construir componentes de software.
import java.image.*;
    Procesa imágenes.
import java.image.renderable.*;
    Renderiza imágenes de forma independiente.
import javax.swing.*;
    Es la librería de componentes visuales mas nueva que proporciona java,
    usando esta librería se podrán construir los tres tipos de programas o
    aplicaciones que son JFrame, JWindow y JApplet.
import javax.swing.event.*;
```

### Definición de las Clases RIG y Edit\_RIG

```
public class EditRIG extends javax.swing.JFrame
```

De donde *JFrame* permite crear ventanas de y se apoya de las siguientes declaraciones de variables definidas en Java:

<i>JFileChooser</i>	Manejador de archivos
<i>JEditorPane</i>	Panel editor de texto
<i>Panel</i>	Superposición de pantallas
<i>JTextField</i>	Cajas de texto
<i>JScrollPane</i>	Secciones de pantalla
<i>Jpanel</i>	Panel de estado
<i>JMenuBar</i>	Barra de menús
<i>Jmenu</i>	Menús principales
<i>JmenuItem</i>	Item del menú

<i>ButtonGroup</i>	Conjunto de botones de la barra de tareas
<i>JsplitPane</i>	Separador de paneles
<i>Jlabel</i>	Panel de mensajes
<i>Jtoolbar</i>	Barra de botones
<i>Jbutton</i>	Botones
<i>Choice</i>	Combos
<i>Jlist</i>	Listas

Y Variables de tipo *Vector*, *String*, *Boolean*, *Int*, *Char*

### 5.3 CLASE EDIT RIG (AGENTE BDATOS)

Desarrolla las funciones propias para la administración de la base de datos, siguiendo el orden presentado en el diseño del entorno con el usuario o interfaz gráfica de usuario.

Main:

```
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new EditRIG().setVisible(true);
        }
    });
}
```

Carga de inicio de programa:

```
public EditRIG() {
    initComponents();           Crear el entorno de desarrollo
    setLocation((1024-600)/2,(768-400)/2); Ubicar la ventana
    ConectarBase();           Conecta la base de datos
    CategoriaCargar();         Cargar categorías
    ImagenCargar();           Cargar imágenes
    UsuarioCargar();          Cargar usuarios
    PalabraGenericaCargar();   Cargar palabras genéricas
}
```

#### 5.3.1 Principal.

Es la pantalla de presentación del prototipo y establece el acceso al módulo de Recuperación de Información Gráfica RIG.

**private void jButton102ActionPerformed:** Establece la conexión con el módulo de administración de la base de datos Editor RIG

### 5.3.2 Categorías.

Operaciones de crear, modificar o eliminar categorías y asignar las imágenes de una categoría, que utiliza las siguientes funciones:

***private void CategoriaCargar():*** Carga el listado de las categorías

***private void CategoriaCrear():*** Crea nuevas categorías

***private void CategoriaModificar():*** Modificar el nombre de las categorías existentes

***private void CategoriaEliminar():*** Elimina una categoría

***private void ImagenCargar():*** Carga el listado de las imágenes

***private void CategoriImagenCrear():*** Crea nuevas imágenes y su relación a una categoría.

***private void CategoriImagenCargar():*** Carga las imágenes asociadas a una categoría seleccionada.

***private void CategoriImagenEliminar():*** Elimina imágenes de una categoría.

### 5.3.3 Palabras genéricas, derivadas y sinónimas.

Esta sección crear las palabras genéricas o atributos de imagen, sus sinónimas y relacionadas, proporcionadas por el usuario según correspondan a la descripción o característica de imágenes:

***private void PalabraGenericaCargar():*** Carga el listado de las palabras genéricas

***private void PalabraGenericaCrear():*** Crea nuevas palabras genéricas

***private void PalabraGenericaModificar():*** Modifica palabras genéricas

***private void ImagenPalabraEliminar():*** Elimina palabras genéricas

***private void PalabraRelacionadaCargar():*** Carga el listado de las palabras relacionadas

***private void PalabraRelacionadaCrear():*** Crea palabras relacionadas a una genérica existente

***private void PalabraRelacionadaModificar():*** Modifica palabras relacionadas



***private void PalabraRelacionadaEliminar():*** Elimina palabras relacionadas

***private void PalabraSinonimaCargar():*** Carga el listado de las palabras sinónimas de las palabras genéricas

***private void PalabraSinonimaCrear():*** Crea palabras sinónimas de una palabra genérica

***private void PalabraSinonimaModificar():*** Modifica palabras sinónimas

***private void PalabraSinonimaEliminar():*** Elimina palabras sinónimas

### **5.3.4 Imágenes y atributos.**

Esta pestaña se encarga de todas las operaciones relacionadas con la imágenes y sus atributos:

***private void ImagenModificar():*** Modifica el nombre imágenes

***private void ImagenEliminar():*** Elimina imágenes

***private void ImagenRutaCrear():*** Establece la ruta con el archivo de la imagen.

***private void PalabraGenericaCargar():*** Carga el listado las palabras genéricas

***private void ImagenPalabraCargar():*** Carga el listado de los atributos de la imagen seleccionada

***private void ImagenPalabraEliminar():*** Elimina el atributo seleccionado de una imagen

### **5.3.5 Reinicio de pesos.**

Con fines demostrativos y experimentales esta sección reinicia los valores de los pesos de los perfiles de trabajo, de las categorías y de las imágenes de un usuario:

***private void UsuarioCargar():*** Carga el listado de usuarios

***private void PerfilUsuarioCargar():*** Carga el listado de los perfiles de trabajo del usuario seleccionado

***private void PerfilUsuarioReinicio():*** Cambia los valores de los pesos de los perfiles de trabajo del usuario seleccionado a cero

***private void CategoriaCargar():*** Carga el listado de las categorías del perfil de trabajo seleccionado

***private void CategoriaReinicio():*** Cambia los valores de los pesos de las categorías del perfil de trabajo seleccionado a cero

***private void CategorialmagenCargar():*** Carga el listado de las imágenes relacionadas con la categoría seleccionada

***private void CategorialmagenReinicio():*** Cambia los valores de los pesos de las imágenes de la categoría seleccionada a cero

## 5.4 CLASE RIG (AGENTE RECUPERACIÓN DE INFORMACIÓN GRÁFICA)

Main:

```
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new RIG().setVisible(true);
        }
    });
}
```

Carga de inicio de programa:

```
public RIG() {
    initComponents();           Crear el entorno de desarrollo
    ConectarBase();           Conecta la base de datos
    UsuarioCargar();          Cargar usuarios
    PerfilCargar();           Cargar perfiles
    CategoriaCargar();        Cargar categorías
    setLocation((1024-600)/2,(768-400)/2); Ubicar la ventana
    jTabbedPane1.setEnabledAt(1,false); Deshabilita la pestaña 1
    jTabbedPane1.setEnabledAt(2,false); Deshabilita la pestaña 2
    jTabbedPane1.setEnabledAt(4,false); Deshabilita la pestaña 4
}
```

### 5.4.1 Principal.

Sección de presentación del prototipo, selección usuarios y del perfil de trabajo, establece vínculo con el administrador de la base de datos Editor RIG:

***private void UsuarioCargar():*** Carga el listado de usuarios

***private void PerfilUsuarioCargar():*** Carga el listado de perfiles

***private void jButton101ActionPerformed:*** Vincula al editor de texto y carga en memoria el perfil de trabajo seleccionado por el usuario

***private void jButton102ActionPerformed:*** Establece vínculo con el Editor RIG

***private void HistorialCrear():*** Crea el historial del usuario seleccionado como resultado de *iniciar* la ejecución del aplicativo

***private void EscucharPortapapeles():*** Inicia el aplicativo en ejecución automática desde cualquier programa que acepte formatos gráficos.

#### **5.4.2 Procesamiento de texto.**

Captura del texto base por inserción o por comandos de portapapeles (copiar y pegar) y ejecuta el procesamiento de texto:

***ProcesoMinusculas():*** Cambia el párrafo a palabras en minúsculas

***ProcesoTildadas():*** Cambia vocales tildadas por no tildadas

***ProcesoDividir():*** Divide en palabras el párrafo

***ProcesoVacías():*** Elimina palabras vacías

***ProcesoLematizacion():*** Determina palabras genéricas a partir de palabras relacionadas

***ProcesoTesauros():*** Determina palabras genéricas a partir de palabras sinónimas

***ProcesoPesos():*** Calcula pesos de las palabras

***ProcesoEliminar():*** Elimina palabras repetidas

***ProcesoResultado():*** Presenta el resultado del procesamiento del texto

#### **5.4.3 Resultados de recuperación.**

Presenta los resultados de la recuperación. Genera un listado de imágenes preseleccionadas ordenadas descendientemente de acuerdo al resultado de la similitud entre las palabras relevantes del texto base y los atributos de las imágenes:

***private void ProcesoResultado():*** Determina que imágenes se mostraran como existas y que palabras se mostraran como palabras relevantes

***class Imagenes extends JFrame:*** Esta clase permite mostrar la imagen seleccionada de la lista

***private void ImagenPalabraCargar():*** Carga el listado de los atributos de la imagen seleccionada

***private void ImagenPalabraCrear():*** Crea nuevos atributos de imagen desde los resultado del procesamiento de texto

***private void ImagenPalabraEliminar():*** Elimina atributos de la imagen seleccionada

#### **5.4.4 Configuración de perfiles.**

Crea o elimina perfiles de usuario, agrega o eliminar perfiles de trabajo asociados a un usuario y agregar o eliminar categorías relacionadas a un perfil de trabajo:

***private void UsuarioCargar():*** Carga el listado de usuarios

***private void UsuarioCrear():*** Crea usuarios

***private void UsuarioEliminar():*** Elimina usuarios

***private void PerfilUsuarioCargar():*** Carga listado de perfiles de usuario

***private void PerfilUsuarioCrear():*** Crea perfiles de usuario

***private void PerfilUsuarioEliminar():*** Elimina perfiles de usuario

***private void PerfilCargar():*** Carga el listado de los perfiles de trabajo

***private void PerfilCrear():*** Crea perfiles de trabajo

***private void CategoriaPerfilCargar():*** Carga categorías

***private void CategoriaPerfilCrear():*** Crea categorías

***private void CategoriaPerfilEliminar():*** Elimina categorías

#### **5.4.5 Historial.**

Muestra los historiales de las sesiones de trabajo realizadas por los usuarios seleccionados:

***private void UsuarioCargar():*** Carga los usuarios

***private void HistorialCargar():*** Carga el historial del usuario seleccionado

## **5.5 DESARROLLO DE LA INTERFAZ CON EL USUARIO**

Genera el entorno de trabajo del prototipo que se dispone ventanas y en pestañas para los diferentes procesos:

*private void initComponents():* Entorno de trabajo

## **5.6 ARCHIVOS ADJUNTOS**

Este capítulo esta acompañado de los archivos del código fuente del sistema y de la base de datos, del archivo ejecutable de instalación del prototipo y de un archivo pdf con el presente documento.

Para la ejecución del programa debe instalar previamente el JDK 1,4 (Java Developer Kit), incluido en los instaladores del CD adjunto.

Para la ejecución del programa desde el código fuente debe incluir el contenido de la carpeta "código fuente" en C:\RIG, carpeta incluida en el CD adjunto.

La base de datos esta cargada con 480 imágenes y alrededor de 1.500 palabras entre genéricas relacionadas y sinónimas.

## CONCLUSIONES

El trabajo expone las principales consideraciones sobre programación orientada a agentes, su evolución y conformación, técnicas y posibilidades de implementación, tomadas desde la base de la programación orientada a objetos como plataforma conceptual, su transformación y ventajas comparativas para mejores productos capaces de aprender de su experiencia y de los invaluable aportes del usuario, así como las principales herramientas para interpretar desde un texto las palabras representativas o relevantes del contenido y así asociarlas a un grupo de imágenes preseleccionadas que toman diferentes posiciones frente al mismo texto en la medida que los resultados de la recuperación sean exitosos ("aprendizaje").

Para esto se expone igualmente las teorías de recuperación de información que se ajustan específicamente a la recuperación de información gráficas que le da un componente adicional como lo es la manipulación de bases de imágenes o colecciones para su clasificación y organización de manera que le permita al usuario hacer uso de ellas de forma eficiente y eficaz.

Y la personalización, como componente que no puede hacer falta, que beneficia tanto a los usuarios como al sistema en la agilidad y certeza de los resultados, ya que utiliza en cada búsqueda sólo aquel grupo de imágenes que predetermine el perfil de trabajo.

La recuperación de información en colecciones digitales o bancos de imágenes son un componente fundamental, junto con los servicios e interfaces, de los complejos sistemas de apoyo a la investigación y aprendizaje colaborativo, componentes esenciales en todo programa moderno de educación e investigación, en la búsqueda de mecanismos para construir y explorar colecciones digitales.

El proyecto tiene como resultado un prototipo de sistema de recuperación de información gráfica basado en la metodología MAS-CommonKADS, para ofrecer un aplicativo funcional que permita incorporar imágenes según sus contenidos, además de que "aprenda" desde los resultados de la recuperación y desde el concepto gráfico de los usuarios.

El prototipo se basó en un agente de software para la recuperación de la información gráfica, para lo que interviene la determinación de los perfiles de usuario y de trabajo, un procesamiento de texto, la asignación de pesos para estimar el aprendizaje y una base de datos, desarrollados en el lenguaje de programación Java.

El principal aporte esta en la integración de un modelo orientado a agentes, las herramientas de recuperación de información, el procesamiento de textos y la búsqueda y manipulación de la base de datos de imágenes y su información.

Como condición del agente está el aprendizaje o experiencia entre lo que interviene y cabe destacar:

- El prototipo permite aprender de los resultados de la recuperación, además de que posibilita replantear el texto base de búsqueda para enriquecer con palabras relevantes más precisas la base de datos.
- La visión/opinión del usuario a través de su interacción y concepto gráfico proporcionan mayor exactitud al agente.
- A mayor interacciones se precisa o afina los resultados.
- La dependencia de contar con una base cada vez más completa de palabras genéricas y sus relacionadas y sinónimas para mejorar los resultados de recuperación, componente importante que sin embargo no dependen del prototipo como sistema y si incide de forma significativa en los resultados, debido a que se depende de la información y de las variaciones lingüísticas que son complicadas de preestablecer.

Los resultados son del todo exitosos ya que las pruebas mostraron que como resultado de la interacción entre el sistema y el usuario y los pesos que van tomando palabras, imágenes, categorías y perfiles de trabajo, los resultado son cada vez más precisos y van cambiando de acuerdo a los intereses de búsqueda.

El enfoque y el sistema experimental desarrollado posee la ventaja de evitar el gran sobreflujo de información innecesaria ya que la relevancia de la palabras producto del procesamiento sólo trabajan con un grupo predelimitado por sus características, trabajando únicamente con sus relacionadas.

## **RECOMENDACIONES**

Las perspectivas están orientadas desde el mismo prototipo y desde el apoyo que puede ofrecer para conformar otros aplicativos:

Desde el prototipo como trabajo futuro se puede incluir en la implementación de los procedimientos de tesauros y lematizaciones la interpretación semántica de las palabras orientado al procesamiento del lenguaje natural, que permitiría hacer mas exhaustivo la recuperación de información gráfica, este trabajo toma dichos procesos en su forma básica como lo es que a palabras genéricas se asocie su sinónimas y relacionadas, que incluiría significativamente aumentar el rango de palabras indexadas. Adicionalmente permitir mejorar la manipulación de imágenes en cuanto a su presentación, selección e incorporación al editor de trabajo.

En cuanto a la posibilidad que ofrece el prototipo para otras aplicaciones se encuentra múltiples alternativas entre las que se puede considerar búsqueda de catálogos, bibliotecas digitas, revistas, directorios, y cualquier otro uso que implique recuperación de información desde información de texto asociada a los archivos objeto de búsqueda.

Por último todas las mejoras que desde el diseño se puedan incorporar en cuanto a la accesibilidad, tiempo de respuesta, fácil manejo, manipulación de imágenes y entornos amigables.



## **BIBLIOGRAFÍA**

ARROLLO MUÑOZ, Ana. Gestión del Conocimiento. Departamento de consultoría tecnológica. Fundación Robotiker. Revista Robotiker.

BAEZA-YATES, R. y RIBEIRO-NETO, B. Modern Information Retrieval. Addison-Wesley, Essex, 1999.

BELKIN, N.J.; CROFT, W.B. Retrieval techniques. Annual Review of Information Science and Technology, 22, p. 109-145, 1987.

BOTTI, V., CARRASCOSA, C., JULIAN, V., SOLER, J. The ARTIS Agent Architecture: Modelling Agents in Hard Real-Time Environments. Proceedings of the MAAMAW'99. Lecture Notes In Computer Science, vol. 1647. Springer-Verlag (pag. 63-76), Valencia 1999. ISBN 3-540-66281-2.

BRADSHAW, J. (ed.) "Software Agents", AAAI Press/ The MIT Press, 1997.

C. J. Van Rijsbergen. Information retrieval. Second Edition. Butter Worths, Lon-don (U.K.), 1979.

CHRISTOPHER, Fox. Lexical Analysis and Stoplists. En W.B. Frakes y R. Baeza-Yates. Information retrieval, data structures and algorithms. New Jersey, London, etc.: Prentice-Hall, 1992.

CORNELL UNIVERSITY. Library Gateway. Llevando de la Teoría a la Práctica. Tutorial de Digitalización de Imágenes.

FLANAGAN, David. Java en pocas palabras. Referencia al instante. Mc Graw Hill. México 1998.

FRANKLIN, S., GRAESSER, A. "Is it an Agent or Just a Program? A Taxonomy for Autonomous Agents". In Proc. Of the 3thd Int. Workshop on Agent Theories, Architectures, and Languages. New York: Springer-Verlag, 1996.

FROST, R. Bases de Datos y Sistemas Expertos. Ingeniería del Conocimiento. Madris Díaz de Santos, 1989.

GARCÍA CABALLERO, Ricardo. Servicio de Documentación Gráfica. Everes S. A. Nuevas Tecnologías al Servicio de la Información Gráfica. FESABID.

GLASSER, N. Contribution to Knowledge Modelling in a Multi-Agent Framework (the coMoMAS Approach). PhD thesis, L'Unibersité Henry Poincaré, Nancy I, France, November 1996.

GROSSE, A., "Position Paper", <http://www.AgentLink.org> DAI Lab. 1995.

HARMAN, D. Ranking Algorithms. //Frakes, W.B.; Baeza-Yates, R. Information retrieval: Data Structures and Algorithms. Prentice-Hall, Englewood Cliffs (NJ), pp. 363-392, 1992.

HIRTLE, Peter. Moving Theory into Practice: Digital Imaging for Libraries and Archives.

HOOPER, R. S. Indexer consistency tests-origin, measurements, results and utilization. Bethesda, MD, 1965.

HUHNS, M., SINGH, M. P.: Readings in Agents. Readings in Agents. Chapter 1, 1 - 24 1998.

IGLESIAS FERNÁNDEZ, Carlos Ángel. Definición de una metodología para el desarrollo de sistemas multiagente MAS-CommonKADS. Enero 1998.

JONSON, James L. Bases de Datos. Modelos, lenguajes, diseño. OXFORD UNIVERSITY PRESS. Mexico 1997.

LIEBERMAN, Henry. Aria: An Agent for Annotating and Retrieving Images. MIT Media Lab.

LÓPEZ YEPES, Alfonso. Revista. Cuadernos de Documentación Multimedia Número 3. Fototecas Digitales en Prensa: formatos gráficos, entornos y sistemas de información. Junio 1994.

NWANA, H. S.: Software Agents: An Overview. Intelligent Nwana, H. S.: Software Agents: An Overview. Intelligent Systems Research. AA&T, 1996.

P. P. DE LA CRUZ, Rui Alexandre. Perfiles de Usuario: En la senda de la personalización. Informe Técnico. Departamento de Informática y Automática. Universidad de Salamanca. Enero 2003.

PRESSMAN, Roger S. Ingeniería del Software. Cuarta Edición. Editorial McGrawHill. 1997.

R. DE HOOG, R. MARTIL, B. WIELINGA, R. TAYLOR, C. Bright, and W. van de Velde. The CommonKADS model set. ESPRIT Project P5248 KADS-II/M1/DM..1b/UvA/018/5.0, University of Amsterdam, Lloyd's Register, Touche Ross Management Consultants & Free University of Brussels, Dec. 1993.

RUSSELL, S., NORVIG, P. "Artificial Intelligence: A Modern Approach", New York, Prentice-Hall. 1995.

SALTON, G. Automatic Information Organization and Retrieval. McGraw-Hill, N.Y.,1968.

SALTON, G. y McGill, M. J.. Introduction to Modern Information Retrieval. McGraw-Hill, Inc., 1983.

SALTON, G.; BUCKLEY, C. Term-Weighting Approaches in Automatic Text Retrieval. En: Information Processing and Management, 24(5), 513-523, 1988.

SHOHAM, Y., "Agent-oriented programming". Artificial Intelligence, Vol. 60, Nº. 1, pp. 51-92, March 1993.

WHITTEN, Jeffrey L. Análisis y Diseño de Sistemas de Información. Tercera Edición. Editorial Mc GrawHill. 1996.

WOOLDRIDGE, M. and JENNINGS, N. R.: Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10(2):115–152, 1995.

ZAZO, Ángel, C. Figuerola, J. L. Alonso, R. Gómez. Recuperación de Información utilizando el modelo vectorial. Participación en el taller CLEF-2001. Technical Report DPTOIA-IT-2002-006. mayo, 2002.