



Universidad Tecnológica de la Mixteca.

**Diseño e Implementación
de una Sociedad de Agentes Inteligentes
Recolectores de Información.**

T E S I S

que para obtener el título de:

Ingeniero en Computación

p r e s e n t a

Hilda Caballero Barbosa.

Huajuapam de León, Oax.

Verano de 1998.

Indice.

Capítulo 1. Introducción	1
1.1 Motivación.	1
1.2 Antecedentes.	3
1.2.1 Inteligencia Artificial Basada en el Conocimiento.	3
1.2.2 Inteligencia Artificial Basada en el Comportamiento.	5
1.2.3. Perspectiva del Trabajo.	7
1.3 Esquema	11
Capítulo 2. Agentes Inteligentes.	13
2.1 Agentes Inteligentes Basados en el Conocimiento y Agentes Inteligentes Basados en el Comportamiento.	13
2.2 Definiciones de agentes.	14
2.2.1 Definición de Maes.	14
2.2.2 Definición de Wooldridge-Jennings	16
2.3 Clasificación de Agentes.	18
2.3.1 Clasificación según autonomía.	18
2.3.2. Clasificación según actividad.	19
2.4 Diseño de Agentes.	20
2.4.1 Diseño ascendente	20
2.4.2 Componentes del agente.	21
2.4.3 Agentes individuales.	22
2.4.4. “Sociedades” de Agentes	23
2.5 Enfoque grupal de los Agentes Inteligentes Basados en el Comportamiento.	24
2.5.1 Colonias de hormigas.	26
2.5.2 Métodos de comunicación entre agentes.	29

2.5.3 Resultado Sinérgico.	31
Capítulo 3. Una “Sociedad ” de Agentes Inteligentes Recolectores de Información.	33
3.1 Descripción del problema.	33
3.1.1 Propuesta de solución.	36
3.1.2 Metas de la instrumentación.	36
3.1.3 Diseño.	37
3.1.3.1 Componentes del agente.	38
3.1.3.2 Agentes individuales.	42
3.1.3.3 “Sociedad” de agentes-hormiga.	44
3.1.3.4 Agente buscador de sinónimos.	45
3.1.4 Implementación.	47
3.1.4.1 Elección de plataforma de trabajo.	48
3.2 Evaluación del sistema.	52
3.2.1 Desempeño grupal.	52
3.2.2 Desempeño de un individuo.	59
3.3 Comparación del sistema con algunos métodos de búsqueda tradicionales.	62
3.3.1 Resultados de la comparación.	63
3.4 Trabajos relacionados.	71
3.5 Trabajo futuro.	78
Capítulo 4. Conclusiones.	80
Referencias.	84
Apéndice A. Especificación de la clase agente-hormiga y del agente buscador de sinónimos.	95
Apéndice B. Guía del usuario.	99

Capítulo 1.

Introducción.

1.1 Motivación.

Es un hecho que al hombre moderno le es indispensable procesar y manipular información. "Quien tiene la Información tiene el Poder" lejos de ser una frase vana es una realidad de nuestro tiempo. Le es necesario a cualquier persona disponer en todo momento de información actualizada: Así como un hombre de negocios necesita conocer las últimas cotizaciones en la Bolsa de Valores para decidir la acción que tomarán sus próximas inversiones, un joven de 15 años requiere consultar en la biblioteca la información pertinente a su tarea del día siguiente. En ambos casos se involucra a una gran cantidad de datos y para cada usuario, los resultados de su búsqueda podrían ser de gran relevancia. Sin duda la computadora ha revolucionado la manera de tratar con tales volúmenes de datos, con el surgimiento de las Bases de Datos (BD) a finales de los años sesenta. "Una BD es un conjunto de datos almacenados de forma integrada y compartida" (Almeida 1992), en donde idealmente se elimina parcial o totalmente cualquier redundancia y se tiene información consistente. Sin embargo, sabemos que las BD ideales suelen no presentarse y que las herramientas de búsqueda en BD no satisfacen las necesidades del usuario, quien en múltiples ocasiones no encuentra la información relevante en sus consultas en las BD. Las herramientas actuales de búsqueda como dBase, Access, y otras similares, suelen no trabajar adecuadamente si existen fallas en la integridad de la BD. Los resultados de las búsquedas por otro lado, en ocasiones presentan información no relacionada con lo que el usuario espera. Es común también, que no se explore la BD con criterios flexibles de modo que información que pudiera ser valiosa para el usuario no le sea mostrada.

El presente trabajo está orientado al problema de la Búsqueda y Recuperación de Información desde una perspectiva de la Inteligencia Artificial Basada en el Comportamiento. **El objetivo del trabajo es la aplicación de una metodología novedosa en el diseño e implementación de entidades computacionales “inteligentes”, conocidas como Agentes, capaces de contender con ambientes imprevisibles y complejos.**

Una BD puede contener información incompleta, inconsistente, con redundancia, que a pesar de ser casos no deseados en una BD ideal, suelen presentarse en el mundo real y son el motivo de que a una BD se le considere un ambiente complejo. Se trata también de un ambiente imprevisible porque no es posible predecir el próximo estado del medio ambiente a partir de su estado actual así como también no es posible especificar la próxima acción que realizará el agente (Russell & Norvig 1995; Guerra 1997).

Los métodos de búsqueda tradicionales tales como tablas hash, búsqueda secuencial, búsqueda binaria, ó los métodos empleados por la Inteligencia Artificial (IA) como árboles de búsqueda, método de generación y prueba, métodos heurísticos, entre otros (Ver Winston 1989 para profundizar en tales métodos de búsqueda) son poco flexibles para responder exitosamente ante diversas situaciones, no obtienen resultados si los datos son imprecisos, están incompletos o si se pretende realizar búsquedas más versátiles (empleando sinónimos, o a través de la consulta al usuario, por ejemplo). Algunos de los métodos de búsqueda anteriores necesitan de una representación en memoria del espacio de búsqueda resultando costosos en carga computacional.

Mientras que la metodología que se empleará en este trabajo ofrece una propuesta innovadora para realizar búsquedas y recuperación de información, permite explotar una BD y pretende aprovecharla al máximo. La metodología persigue la instrumentación de un buscador y recuperador de información con “comportamiento inteligente”, en el siguiente sentido (Muñoz 1997):

- a) Flexibilidad para contender con diversas situaciones.
- b) Oportunismo para sacar la mayor ventaja y utilidad ante diversas situaciones.
- c) Capacidad para contender con información ambigua, imprecisa e incluso contradictoria.

- d) Capacidad para interactuar exitosamente con entornos complejos e imprevisibles.
- e) Respuesta en tiempo real.

Este comportamiento inteligente al que se ha limitado nuestro estudio, es frecuentemente esperado en sistemas artificiales, como son los robots móviles autónomos, los agentes asistentes o tutores, los buscadores en grandes bases de datos y los sistemas reactivos y basados en el comportamiento (Nilsson 1995; Muñoz 1997).

Con este trabajo se pretende también contribuir al entendimiento de los sistemas autónomos que trabajan en ambientes reales y complejos.

Otro objetivo que se persigue es el estudio de sociedades artificiales mediante la experimentación. El objetivo de esta experimentación es probar la recolección grupal.

1.2 Antecedentes.

Uno de los principales objetivos de la IA ha sido el estudio del fenómeno de la "inteligencia". La fundación de la IA como tal, ocurrió durante la conferencia de Dartmouth en 1956 y mediante la publicación del libro *Computers & Thought* en 1963. La conferencia de Dartmouth fue denominada y organizada en su mayor parte por John McCarthy como el "Proyecto de Verano de Investigación de Dartmouth sobre Inteligencia Artificial". Podemos clasificar a la IA desde dos perspectivas :

- IA Clásica o IA Basada en el Conocimiento (IABCn).
- IA Basada en el Comportamiento (IABCm) .

1.2.1 Inteligencia Artificial Basada en el Conocimiento.

Tradicionalmente, la Inteligencia Artificial Clásica ha manejado la representación del conocimiento y se ha encargado de modelar y de construir sistemas que "conocen" sobre algún dominio del problema. Estos sistemas pueden modelar el dominio y responder preguntas sobre éste dominio del

problema. La IA Clásica tiene dos objetivos vistos desde enfoques distintos: Por un lado y desde el punto de vista de la ingeniería, trata de resolver problemas y, por otro lado, desde el punto de vista científico, trata de explicar varios tipos de inteligencia (Winston 1989). La IA Clásica ha hecho énfasis en la manipulación abstracta de símbolos, es decir, se basa en la hipótesis del sistema de símbolos. Los símbolos pueden ser objetos individuales, propiedades, conceptos, deseos, emociones, colores, bibliotecas, o moléculas. Los símbolos son entidades básicas para realizar la abstracción del mundo real. La hipótesis del sistema de símbolos establece que la inteligencia opera en un sistema de símbolos el cual podrá organizarse para exhibir inteligencia (Newell & Simon 1976). La IABCn representa un problema por medio de la correspondencia semántica entre símbolos y considera necesario abstraer y representar el mundo físico por medio de tales correspondencias.

Los sistemas modelados por la IABCn son aislados y avanzados, muestran “experticia” tanto en "profundidad" como en "amplitud". Son "sistemas cerrados" debido a que su única interacción con el ambiente se realiza a través del usuario y aunque los sistemas tienen el conocimiento sobre el problema y lo resuelven, no interactúan directamente con el dominio de éste. Estos sistemas no tienen restricciones de tiempo para dar soluciones, sólo resuelven un problema a la vez y es indispensable que el dominio del problema sea estable y estático. Se basa en la construcción descendente (Top-down), parte de un alto nivel cognoscitivo con la representación del dominio del problema en una Base de Conocimientos que consta de estructuras de conocimiento estáticas para determinar la solución del problema, las cuales modelan la “experticia” del sistema. El problema y las situaciones que se presentan se trasladan a situaciones simbólicas. Se cuenta además con un sistema central encargado de proporcionar la solución al problema (Maes 1993).

Durante mucho tiempo la IABCn se dedicó al estudio de robots que se desempeñaban en ambientes "ingenierados" , es decir, en ambientes modelados y poco flexibles al cambio. Se tenían mundos simples y situaciones controladas (Brooks 1991a). Para la IABCn las representaciones adecuadas son la clave de una buena resolución de problemas, una vez que se ha descrito un problema mediante una representación apropiada el problema está casi resuelto (Winston 1989).

Ejemplos de sistemas resultantes de esta línea de investigación son los programas de diagnóstico médico y los juegos de ajedrez. Por mencionar sólo algunos tenemos al programa de Weizenbaum llamado ELIZA (Mauldin 1994) que demostró ser un programa simple que podía jugar satisfactoriamente el "Juego de la Imitación" propuesto por Alan Turing (Véase Turing 1950 para mayor información) y que reordenaba unas cuantas "habilidades", respondía preguntas con preguntas, tenía la habilidad de "idear" preguntas con base en la conversación del usuario, quien podía suponer que el programa lo estaba escuchando. Otro digno representante de la IABCn es el programa PARRY de Colby (Mauldin 1994) que simula un comportamiento paranoico, PARRY cambiaba el nivel de la conversación, incluía historias de la Mafia y trataba de mezclarlas al relatarlas, fue tal el éxito de PARRY que ni los siquiátras o computólogos eran capaces de distinguir entre un paciente humano o un "paciente máquina". PARRY fue superior a ELIZA por contar con una "personalidad". Por su parte SHAKEY, un robot móvil basado en la IABCn, operaba en cuartos vacíos, excepto por los grandes bloques de colores que eran los obstáculos alrededor de los cuales se movía y las cuñas que le ayudaban a guiarse a través de los cuartos. Las paredes eran de un color uniforme, cuidadosamente iluminadas, marcando límites con el piso de color más claro. Sin embargo sólo funcionaba debido a que este ambiente estaba cuidadosamente "ingenierado" sin poder desempeñarse en un ambiente más general como el de una oficina (Brooks 1991a). El Robot SHAKEY fue el primer robot móvil desarrollado por el Centro de IA en el Instituto de Investigación de Stanford de 1966 a 1972, SHAKEY usó programas para la percepción, para modelar su ambiente y para actuar en él. Con rutinas de bajo nivel podía realizar movimientos simples y planear rutas. Las acciones intermedias, que aún seguía siendo de bajo nivel, lo ayudaban a realizar tareas más complejas. Y con programas de más alto nivel podía ejecutar planes para conseguir metas dadas por el usuario. El sistema de SHAKEY también se generalizó para almacenar planes para uso futuro (Huetis 1997).

1.2.2 Inteligencia Artificial Basada en el Comportamiento.

A partir de 1984 un grupo de investigadores encabezados por Rodney Brooks, propone conducirse por una nueva línea de investigación, la IABCm. La IABCm trata de modelar y construir sistemas que lleguen a "comportarse"

en algún dominio del problema. Se basa en la hipótesis de la conexión física, la cual establece que para modelar la inteligencia es necesario tener representaciones "aterrizadas" al mundo físico a través de un conjunto de sensores y actuadores.

Los sistemas que modela tienen múltiples capacidades integradas de bajo nivel, como podrían ser en un robot las capacidades de locomoción, de navegación y de recolección de objetos, entre otras. Son sistemas "abiertos" o "situados" que interactúan directamente con su ambiente (Mataric 1994). Están "aterrizados" al dominio del problema a través de sensores y actuadores. Pueden cambiar tal dominio a través de sus acciones, el dominio resulta ser dinámico y comúnmente muy complejo e imprevisible. Los sistemas interactúan en el ambiente con otros sistemas autónomos llamados "agentes" que pueden ser humanos o artificiales. La respuesta a sus percepciones se da en tiempo real. Hacen uso de estructuras activas que son "módulos de comportamiento" que no obedecen a una inicialización de metas o propósitos formulados por el usuario, sino que más bien estas metas son propiedades emergentes que no están definidas explícitamente en alguna estructura interna.

La IABCm se basa en la descomposición de la inteligencia en módulos que generen un comportamiento individual, los cuales coexistan y cooperen haciendo emerger un comportamiento más complejo. Lo anterior permite adicionalmente mejorar la capacidad del sistema sólo al añadir nuevos módulos a éste. Se basa en la construcción ascendente (Bottom-up), parte de niveles con mínima capacidad cognoscitiva hasta lograr incrementalmente comportamientos "inteligentes" complejos, que no se encuentran programados en ningún módulo de forma explícita y que son resultado emergente de la interacción entre los módulos que conforman al sistema y de la interacción del sistema con su medio ambiente. Parte de lo simple a lo complicado.

A la IABCm le conciernen sistemas reactivos con bajos niveles cognoscitivos y altamente dinámicos. Para la IABCm la observación principal es que "el mundo es el mejor modelo de tal sistema" (Brooks 1990, 1991a).

Por referirse a algunos sistemas que operan sobre esta línea de investigación tenemos al robot ALLEN que evitaba colisiones con obstáculos estáticos o en movimiento, para ello contaba con sonares como sensores.

TOM y JERRY son dos robots móviles, físicamente formados por dos carros con tres sensores infrarrojos colocados al frente y uno en la parte de trasera; estos robots evitaban al igual que ALLEN colisiones con obstáculos y se atraían entre ellos, no tenían conocimiento uno del otro y sin embargo emergía entre ellos un comportamiento de persecución (Brooks 1990). Cabe resaltar los trabajos sobre el comportamiento grupal de agentes inspirados en una metáfora de los sistemas biológicos, y a los cuales pertenece nuestro trabajo. Como puede observarse, la IABCm está dedicada al estudio de sistemas que operen en un mundo físico y por lo tanto se enfrenten a los problemas que esto conlleva. Los mundos dinámicos y complejos en donde estos sistemas "habitan" exhiben situaciones imprevistas y variables, por lo que estos sistemas constituyen una propuesta digna de considerarse en la solución a problemas del mundo real.

1.2.3 Perspectiva del Trabajo .

El problema de la Búsqueda y Recuperación de Información es estudiado desde la perspectiva de "agentes inteligentes" inspirados en la metáfora biológica del comportamiento de insectos sociales. Los insectos sociales son mejor ejemplificados por las termitas, varias especies de hormigas y abejas (Wilson 1998b). Nuestro trabajo se inspira en el comportamiento de un hormiguero para resolver colectivamente el problema de búsqueda y recuperación de información y utiliza estrategias de reclutamiento para la recolección de información (Muñoz 1997). La estrategia de reclutamiento entre las hormigas consiste en que, cuando una hormiga encuentra alimento, secreta una sustancia llamada feromona, la cual es utilizada por las hormigas para comunicarse (Wilson 1998a) y para marcar el camino hacia la fuente de alimento; este rastro de feromona conducirá muy probablemente a las otras hormigas a dicha fuente.

Las feromonas son ampliamente usadas para promover la agregación entre los insectos sociales tales como las termitas y hormigas, varios tipos de feromona pueden transmitir varios mensajes requeridos para coordinar las actividades complejas de la colonia. Algunas hormigas dejan una marca de feromona a través de un camino que conduce a una fuente de comida para que otros miembros de la colonia puedan encontrarla. Las feromonas también se usan para indicar la presencia de peligro y juegan un papel importante

también en la atracción sexual. La Feromona es una sustancia química endógena secretada en cantidades diminutas por un organismo para producir una reacción particular de otro organismo de la misma especie; es común entre los insectos y vertebrados pero es desconocida entre los pájaros. Estas sustancias químicas son secretadas por glándulas especiales o están incorporadas en otras sustancias, tales como la orina. Pueden ser esparcidas libremente en el ambiente o depositadas en lugares cuidadosamente elegidos (Wilson 1998c). Entre mayor sea el número de hormigas que refuerzan el camino de feromona menor será el tiempo en que se agotará la fuente de alimento; cuando esto suceda, las hormigas dejarán de recolectar y de secretar feromona, como esta sustancia es volátil el camino formado por ella desaparecerá.

La inspiración biológica del trabajo es pertinente debido a nuestro interés por instrumentar sistemas de bajo costo, de pobres capacidades cognoscitivas y eficientes para resolver colectivamente un problema de cierta complejidad: la búsqueda y recuperación de información, problema existente que demanda de la mejora de métodos y de algoritmos para su resolución. Las habilidades de recolección y de auto-organización que se desea posean nuestros agentes, son pericias probadas en colonias de insectos sociales y que son dignas de estudiarse y de considerarse. Además la auto-organización que se da en estos sistemas muestra que no existe una autoridad central y que la organización se obtiene de forma distribuida, dando mayor robustez y flexibilidad al sistema.

Con una colección de agentes sin conocimiento del mundo en que habitan, sin programación compleja y mediante el uso de mecanismos de comunicación elementales, son capaces de colaborar para resolver el problema que nos ocupa. No hay controladores centrales, ni jefes, ni castas, la gracia y la contribución de este trabajo es que a partir de una programación simple se puede resolver lo no simple, o como lo diría Brooks (Brooks 1990), la búsqueda de la inteligencia bien puede empezar desde el sustrato elemental, de forma 'Bottom-up'.

Resulta interesante estudiar los mecanismos de comunicación que se requieren para conseguir la colaboración, con este sistema se desea resolver cómo es que un agente-hormiga al encontrar información relevante contribuye con los demás.

La gran diversidad de aplicaciones y propuestas es un signo claro de que los agentes de software están llegando a ser importantes. Estas aplicaciones están orientadas en la Inteligencia Artificial Distribuida (IAD), robótica, IA, computación de un objeto distribuido, interacción hombre-máquina, interfaces adaptativas e inteligentes, búsqueda y filtrado inteligente de información, recuperación de información, adquisición de conocimiento, entre otras (Nilsson 1995; Bradshaw 1997).

Nos interesa participar de las principales aplicaciones potenciales de la tecnología de agentes que son:

Solución Cooperativa del problema e Inteligencia Artificial Distribuida (IAD). La IAD ha hecho énfasis en el macro fenómeno (nivel social) en lugar del micro fenómeno (nivel de agente). La IAD se enfoca a cómo un grupo de agentes pueden cooperar para solucionar eficientemente problemas, y cómo las actividades del grupo de agentes pueden ser eficientemente coordinadas. Gracias a la computación distribuida los agentes de software inherentemente se benefician de los beneficios potenciales en la IAD incluyendo la modularidad, velocidad (debido al paralelismo) y la seguridad (debido a la redundancia) (Mataric 1995; Nwana 1996).

Los agentes que se pretende implementar son agentes colaborativos, la motivación para tener sistemas de agentes colaborativos podría incluir una o varias de las siguientes razones (Nwana 1996):

- Solucionar problemas que son demasiado grandes para un único agente centralizado debido a las limitaciones de la fuente o al riesgo de tener un sistema centralizado,
- Proporcionar soluciones para problemas inherentemente distribuidos,
- Proporcionar soluciones las cuales se extraen de fuentes distribuidas de información, ejemplo: fuentes de información *on-line* distribuida, es natural adoptar una propuesta distribuida y colaborativa,
- Proporcionar soluciones donde la “experticia” es distribuida.
- Incrementar la modularidad (lo cual reduce la complejidad), la velocidad (debido al paralelismo), confiable (debido a la redundancia), flexible (nuevas tareas se componen más fácilmente de

una organización modular) y la reusabilidad en el nivel de conocimientos (de aquí el compartir recursos).

Además se plantea a la tecnología de agentes para dar solución a un problema complejo, las herramientas más poderosas para manejar la complejidad en el desarrollo de software son la modularidad y la abstracción. Los agentes representan una herramienta poderosa para hacer sistemas modulares. Si un dominio del problema es particularmente complejo, grande o imprevisible, como lo es para el problema que nos ocupa, entonces podría ser que la única forma en que se puede solucionar un aspecto particular del problema es a través de un número determinado de componentes modulares especializados. En tales dominios, una propuesta basada en agentes significa que el problema puede ser particionado en un número de componentes más pequeños y más simples, los cuales son más fáciles de desarrollar y mantener, y los cuales son especializados en solucionar los subproblemas. De modo que la solución que se aborda a través de una sociedad de agentes autónomos y cooperativos. Para muchas aplicaciones, esta forma de solucionar un problema puede resultar la más apropiada (Jennings & Wooldridge 1998)

Nuestro sistema multi-agente estará enfocado a la solución del problema de búsqueda y recuperación de información. Dichos agentes son clasificados como agentes de información¹. Un agente de información es un agente que tiene acceso a por lo menos una, y potencialmente a muchas fuentes de información, y es capaz de verificar y manipular información obtenida de estas fuentes de acuerdo a respuestas de *queries* planteados por los usuarios y otros agentes de información. Las fuentes de información pueden ser de muchos tipos, incluyendo, por ejemplo, las bases de datos tradicionales así como otros agentes de información. Para encontrar una solución a un *query* se podría involucrar a un agente que accese a fuentes de información sobre la red (Jennings & Wooldridge 1995). Los agentes de información tienen características variables: pueden ser estáticos o móviles, pueden ser sociales o no cooperativos y pueden o no aprender (Nwana 1996).

Adicionalmente con nuestro trabajo se trata de colaborar con la investigación en agentes de Información distribuidos. Esta investigación hace énfasis en el desarrollo de propuestas para particionar el espacio de la

¹ En el capítulo 2 se da una clasificación más formal de agentes.

búsqueda en un conjunto de búsquedas locales y coordinarlas simultáneamente en términos de controlar y compartir la información. En sistemas de agentes que están lógicamente distribuidos, la partición del espacio de estados del sistema se induce por la descomposición del problema a priori impuesto por el conjunto de agentes. La selección del agente es un proceso iterativo. En los agentes cooperativos en dominios locales, que es el caso que nos ocupa, su contribución potencial resulta en un solución colectiva (Lander 1994). Se da una solución distribuida al problema de la adquisición cooperativa de la información (Oates et. al. 1994).

Del diseño e implementación de nuestra colonia artificial se desea también pueda proporcionar resultados útiles al paradigma *Functionally-Accurate, Cooperative (FA/C)* que se emplea en la solución distribuida de un problema. Este paradigma es una propuesta para solucionar un problema mediante la organización distribuida entre agentes homogéneos cooperativos. La idea detrás del modelo de FA/C es que los agentes deberían producir resultados tentativos, parciales y basados sólo en la información local y al intercambiar estos resultados entre los agentes (Carver & Lesser 1994). El sistema que proponemos es un sistema multi-agente homogéneo que mediante una organización distribuida colaboran en la recuperación colectiva de información. Cada agente tiene acceso sólo a información local de la BD mediante los sensores de que está dotado.

La tecnología de agentes pretende mejorar la eficiencia en el desarrollo de software. Proporciona herramientas con las cuales construir aplicaciones que anteriormente éramos incapaces de construir. Pero también proporciona mejores medios de conceptualización y/o implementación a una aplicación dada. Uno de los dominios importantes que se caracterizan por ser frecuentemente citado como racionalmente adoptado por la tecnología de agentes es el que considera al sistema como una sociedad de componentes autónomos cooperativos (Jennings & Wooldridge 1998).

1.3 Esquema.

El trabajo está organizado en cuatro capítulos. El primero de ellos se refiere a la motivación del trabajo, se presenta también la perspectiva del trabajo y el esquema del mismo.

El segundo capítulo está dedicado a la Metodologías Basadas en el Comportamiento las cuales son la fundamentación teórica sobre la que descansa el trabajo. Se presenta una disertación sobre el término agente inteligente Basado en el Comportamiento y una clasificación de agentes. Se analizan los problemas que hay que enfrentar en su diseño e implementación. Y finalmente se distinguen las propiedades de auto-organización y cooperación que acontecen en las agrupaciones de agentes.

El tercer capítulo se concentra en el diseño, implementación y evaluación de una "sociedad" de agentes inteligentes dedicados a la tarea de recolectar información. El diseño se basa en la metáfora de los insectos sociales y se presenta como una alternativa para dar solución al problema de forma simple y con bajo costo. Se muestra una comparación del sistema con otros métodos de búsqueda y se estudian además algunos trabajos relacionados con nuestra área de estudio. El tercer capítulo finaliza con las perspectivas futuras que podrían derivarse de nuestra investigación.

El cuarto y último capítulo está referido a las conclusiones derivadas del trabajo.

Capítulo 2.

Agentes Inteligentes.

2.1 Agentes Inteligentes Basados en el Comportamiento y Agentes Inteligentes Basados en el Conocimiento.

En principio es preciso diferenciar entre un Agente Basado en el Conocimiento (ABCn) y un Agente Basado en el Comportamiento (ABCm).

Un ABCn, como es de esperarse, cumple con las características de la IABCn vistas en la introducción del trabajo. Tiene una representación global del mundo en el que se desenvuelve. Esta representación, por medio de símbolos, es una abstracción del mundo real. Toda abstracción implica una reducción de la información así que el agente posee una noción reducida del mundo real y requiere para operar adecuadamente de ciertas condiciones que no alteren su modelo del mundo. Además no interactúa directamente con el mundo. La "inteligencia" en él es un fenómeno establecido, es decir, programado explícitamente.

Un ABCm a su vez satisface las características del paradigma de la IABCm y no cuenta con una representación global del mundo. Un ABCm se encuentra situado en el mundo y no requiere de descripciones abstractas, cuenta con perceptores y efectores adecuados para experimentar e interactuar directamente con el mundo. Su "inteligencia" no está programada explícitamente sino más bien es resultado de las interacciones internas del agente y de las interacciones entre él y su medio ambiente. La "inteligencia" se determina por la conducta total del sistema, es decir, la "inteligencia" emerge (Brooks 1991a).

Se ha dado una descripción genérica de los agentes que aquí se estudian, los Agentes Basados en el Comportamiento, pero para lograr una mejor comprensión del término agente se presenta una serie de definiciones que ha adoptado la comunidad de la IA, para finalizar con la definición que cumpla con las propiedades que debe poseer un agente según lo postulado en el presente trabajo. En adelante se utilizará el término agente para referirse a un ABCm.

2.2 Definiciones de agente.

Es necesario distinguir a un agente del resto de los programas. "De modo genérico un agente es aquel que percibe su entorno mediante sensores y actúa en él mediante efectores" (Russell & Norvig 1995). Los sensores o perceptores aterrizan al agente en su medio ambiente mientras que los actuadores o efectores le permiten al agente interactuar con él. A esta propiedad del agente se le conoce como reactividad (Russell & Norvig 1995; Guerra 1997).

Esta noción no nos es suficiente debido a que un termorregulador podría intuitivamente interpretarse como un agente: percibe la temperatura de un horno y en consecuencia actúa. O bien "... una macro de mi procesador de palabras que percibe una combinación de teclas que he presionado y actúa realizando algunas modificaciones en el formato de mi texto, es también un agente" (Guerra 1997). Estos ejemplos muestran que un agente debe poseer otras propiedades además de la Percepción y Actuación para ser considerado como tal.

2.2.1 Definición de Maes ².

"Los agentes autónomos son agentes computacionales que habitan en algún ambiente dinámico y complejo, percibiendo y actuando en forma autónoma en este ambiente, y lo hacen para cumplir un conjunto de propósitos o tareas para las cuales fueron diseñados".

Analizando esta acepción se tiene que:

² Maes según (Franklin & Graesser 1996)

- Restringe la ubicación del agente a un ambiente dinámico y complejo. La relación entre el agente y su ambiente cobra importancia, el ambiente dota de percepciones al agente y éste efectúa acciones sobre él. Un ambiente puede tener diferentes propiedades que condicionan de alguna manera el diseño del agente (Russell & Norvig 1995; Guerra 1997): Un ambiente es accesible si los perceptores del agente le permiten tener acceso al estado total de su ambiente. Determinista si es posible determinar el siguiente estado a partir del actual y de las acciones elegidas por los agentes. Episódico si el ambiente puede dividirse en episodios de percepción-acción. Dinámico si existe la posibilidad de que el ambiente sufra modificaciones mientras el agente se encuentra en proceso de deliberación, de lo contrario es un ambiente Estático. Y es un ambiente Semidinámico si son las acciones del agente las que cambian con el paso del tiempo. Un ambiente es Discreto si existe un número limitado de percepciones y acciones diferenciables, o de lo contrario es un ambiente Continuo.
- El agente o agentes deben actuar de manera autónoma en el ambiente en que habitan. Un agente será autónomo en la medida en que su conducta esté definida por su propia experiencia y no sea dirigida por otros (McFarland 1995). Debe ser capaz de funcionar satisfactoriamente en una amplia gama de ambientes considerando que se le da el tiempo suficiente para adaptarse (Russell & Norvig 1995).
- Debe actuar en respuesta a lo que ha percibido de su entorno pero regido por propósitos determinados. Esto sugiere que las acciones de un agente están encaminadas hacia el logro de una meta.

De esta definición se destacan dos propiedades de agentes que se relacionan: La **autonomía** y el **cumplimiento de metas**. Una entidad debe ser autónoma para ser verdaderamente inteligente. Una entidad es autónoma si percibe que tiene metas que cumplir, incluyendo cierta clase de metas, y es capaz de elegir entre una variedad de metas que intenta alcanzar. El mayor requerimiento para la autonomía es que la entidad debe tratar de realizar algo, es decir, debe estar orientada a una meta (Covrigaru & Lindsay 1991):

La autonomía tiene a su vez distintas propiedades : El agente es autónomo si es capaz de almacenar y usar energía para iniciar movimiento

bajo situaciones apropiadas del ambiente. Este movimiento debe verse como un evento de procesamiento de información -como respuesta a los estímulos del ambiente- en lugar de un evento meramente físico. El movimiento al que se hace referencia es más bien la interacción flexible y adaptativa que se tenga con el ambiente; la interacción debe ser "robusta", es decir, debe tener la habilidad para "sobrevivir" en diferentes situaciones, se requiere "robustez" para lograr la autosuficiencia que es la habilidad para responder de determinada manera en un ambiente variable sin la intervención de otras entidades con el fin de mantenerse libre de peligro y de permanecer viable e intacto (McFarland 1995). Una propiedad importante de la autonomía es la independencia, es decir, una entidad es autónoma si sus acciones no son controladas o influenciadas en demasía por su ambiente u otras entidades y no requiere de la asistencia o conocimiento de otros, es independiente porque no requiere para su existencia de un conocimiento detallado.

Aún falta una característica importante del concepto de agente que presenta la siguiente definición.

2.2.2 Definición de Wooldridge-Jennings³.

"Un agente es un sistema de cómputo de hardware o más comúnmente de software, que consta de las siguientes propiedades:

- *autonomía: Los agentes operan sin la intervención directa de humanos u otros, y tienen algún tipo de control sobre sus acciones y sobre su estado interno;*
- *sociabilidad⁴: Los agentes interactúan con otros agentes (y posiblemente humanos) vía algún tipo de lenguaje de comunicación con agentes;*
- *reactividad: Los agentes perciben su ambiente, (el cual puede ser el mundo físico, un usuario vía una interfaz gráfica, una colección de otros agentes, el Internet, o tal vez todos estos combinados), y responde oportunamente adaptándose a cambios que ocurran en él;*

³ Wooldridge-Jennings según (Franklin & Graesser 1996)

⁴ Los autores se refieren a agentes estudiados por la IAD y a Sistemas Multi-agentes.

- *iniciativa: Los agentes no simplemente actúan en respuesta a su ambiente, son capaces además de exhibir un comportamiento de propósito directo tomando la iniciativa".*

Aquí se menciona la propiedad de asociación entre agentes y un requerimiento de comunicación adicional. Se trata de explotar las dinámicas de las interacciones locales entre los agentes y el mundo para crear comportamientos globales complejos, estos comportamientos emergen de tales interacciones. El resultado es un sistema multi-agente que involucra estrategias simples de comunicación. La comunicación en el sistema ayuda a lograr la cooperación⁵ entre los agentes, el comportamiento de cada individuo dentro del sistema contribuye a su funcionamiento, se tiene un resultado sinérgico donde el desempeño del sistema es más que la suma de las partes. El funcionamiento de los sistemas Multiagente Basados en el Comportamiento suelen inspirarse en sistemas biológicos, y particularmente en las conductas de insectos sociales, a estos sistemas también se les llama sociedades de agentes⁶.

Para otros investigadores el término agente tiene un significado más específico. Estos investigadores generalmente entienden que un agente es un sistema que, además de tener las propiedades antes identificadas, puede conceptualizarse o implementarse usando conceptos mentales, tales como el conocer, el creer, la intención y la obligación. Uno de estos investigadores llamado Yoav Shoham propuso un nuevo paradigma de programación llamado Programación Orientada a Agentes (POA). La idea principal del paradigma de POA es que los agentes se programan directamente representando sus propiedades en términos de conceptos mentales e intencionales. Esta propuesta considera que podría ser útil usar la postura intencional para programar a las máquinas. La POA puede verse como una especificación del paradigma de la Programación Orientada a Objetos (Para abundar al respecto ver Shoham 1993).

⁵ Utilizamos el término *Cooperación* para referirnos a las conductas que contribuyen a mejorar el desempeño de un grupo de entidades con comportamiento inteligente (Muñoz 1997).

⁶ El término "Sociedad" no sólo significa agrupación, es más bien utilizado por la similitud con las *eccionsociedades biológicas*. Se utiliza en éste trabajo en su acepción biológica, debido a que los sistemas artificiales se han inspirado en las sociedades biológicas tales como insectos sociales, bandadas de aves, bancos de peces.

2.3 Clasificación de agentes.

Al igual que definiciones, existen diferentes clasificaciones de agentes, a continuación se presenta una clasificación que ayuda a ubicar al tipo de agentes que son objeto de estudio en esta tesis.

2.3.1 Clasificación según autonomía.

En una primera clasificación (Guerra 1997) se distingue entre agentes autónomos naturales (seres humanos) y artificiales. Los agentes artificiales pueden ser clasificados a su vez, con base en el medio ambiente en el que actúan. De esta forma podemos establecer las clases de agente robótico, para aquellos agentes artificiales que actúan en el mundo real, y agentes computacionales (softbots) para los que actúan en un ambiente virtual implementado en una computadora. Véase figura 2.1.

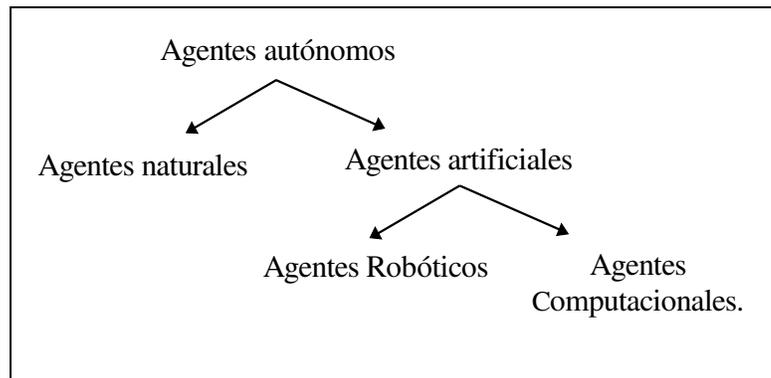


Figura 2.1. Clasificación de Agentes por (Guerra 1997) .

Los agentes que se presentan en este trabajo pertenecen al grupo de softbots. Esta clasificación sin embargo, no dice nada sobre su desempeño, por lo que analizaremos en la sección siguiente otra clasificación propuesta por Guerra en (Guerra 1997).

2.3.2 Clasificación según actividad.

Se propone una clasificación de agente con base en el servicio que estos pueden prestar, un *agente de servicio* es aquel que lleva a cabo tareas bien definidas a petición del usuario. Los llamados agentes de información podrían entrar en esta categoría. Estos agentes de información operan en Internet y pueden ser: agentes guía que se encargan de orientar a las personas que navegan el World Wide Web, agentes constructores de índices de sitios en la red como Lycos, WebCrawler e Infoseek -realizan de manera autónoma una búsqueda masiva en Internet-, agentes localizadores de FAQs los cuales buscan información relevante en los foros de discusión. *Un agente predictivo* provee información o servicios al usuario sin un requerimiento explícito para ello. Su desempeño mejora con base en la retroalimentación con el usuario. Los *agentes cooperativos* hacen énfasis en el nivel social de los agentes o en el nivel multitudinario. Estos agentes solucionan de manera cooperativa problemas como el control de tráfico aéreo, el control de aceleradores de partículas, la administración de plantas generadoras de energía eléctrica y el control de naves espaciales, entre otros. Véase figura 2.2.

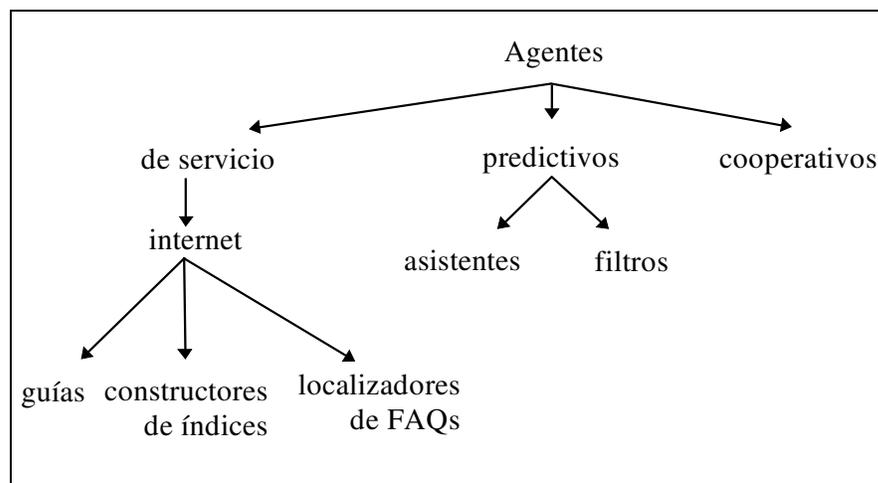


Figura 2.2. Clasificación de agentes por la forma en que prestan su servicio (Guerra 1997) .

2.4 Diseño de agentes.

2.4.1 Diseño ascendente.

Existe un gran interés en el campo de la IABCm por analizar cuáles son los componentes necesarios en la arquitectura del diseño de agentes. No existe una única teoría que nos diga qué es lo que necesitamos conocer para diseñar un agente, Brooks propone una agenda de trabajo a seguir para el desarrollo de la arquitectura de agentes (Brooks 1991b), esta agenda se siguió en el diseño de los agentes artificiales que aquí se proponen.

La descomposición natural de la tarea se da en tres niveles:

- a) Micro nivel: Estudia el cómo se mantienen los diferentes lazos entre el agente y su ambiente para alguna tarea en particular. En este nivel hay que determinar las formas de percepción que son necesarias, y las relaciones que existen entre la percepción, el estado interno y la acción, es decir, la manera en que se especifica o describe el comportamiento. La agenda de trabajo de Brooks indica que lo primero que se requiere analizar son los componentes del agente, los perceptores y efectores con los que debe contar para poder interactuar autónomamente con su medio ambiente.
- b) Macro nivel: Estudia el cómo muchas percepciones del micro nivel y comportamientos se integran en un agente individual. En este nivel se trata de construir al agente artificial de modo que cumpla con todos sus comportamientos individuales requeridos y que se incrementen exhibiendo comportamientos complejos que puedan emerger de reflejos simples. En este nivel es preciso dotar al agente de un mecanismo adecuado de selección de comportamientos que le permita cumplir con la meta por la cual se guía.
- c) Multinivel: Este nivel estudia la forma en que interactúan una multitud de individuos o agentes. Se ocupa de las relaciones entre los comportamientos individuales, la cantidad y el tipo de comunicación entre agentes, la forma en que múltiples individuos reaccionan al ambiente, y los patrones resultantes del comportamiento y su impacto sobre este, lo cual no ocurre en el caso de individuos aislados. En este nivel se estudia la forma de comunicación entre los individuos para

hacer emerger un comportamiento grupal complejo, se diseñan "sociedades" de agentes que exhiban comportamientos colectivos y de cooperación.

Resumiendo la agenda de trabajo: Primero se debe diseñar e implementar los componentes necesarios del agente -perceptores y efectores- estableciendo sus conductas básicas. En segundo término debe diseñarse un mecanismo de selección para permitirle al agente elegir la conducta adecuada en cada momento. Finalmente deben instrumentarse mecanismos de comunicación para que un agente pueda agruparse con otros agentes idénticos o similares. En las secciones siguientes se muestra este proceso más a detalle.

2.4.2 Componentes del agente.

Cualquier agente artificial debe ser capaz de interactuar con su ambiente. Los aspectos relevantes en este nivel son (Brooks 1990, 1991a, 1991b; Nilsson 1995):

- a) Convergencia: Se debe demostrar o probar que un agente está programado en tal forma que su comportamiento externo alcanzará una tarea en particular con éxito. Bajo algunas condiciones y limitaciones iniciales del mundo donde se colocará al agente, esta seguirá un camino en particular en lugar de diverger y "perdersé".
- b) Complejidad: El agente debe contender con la complejidad del mundo, para lograrlo debe examinar los aspectos relevantes de las sensaciones recibidas.
- c) Síntesis: Para una tarea en particular, el programa debe funcionar de tal forma que el agente demuestre convergencia.

El agente interactúa con el mundo, lo percibe y actúa en él, cambia su propia percepción, y tal vez cambia al mundo mismo. El agente requiere procesar sólo aspectos del mundo que son relevantes a su tarea. En este nivel se propone la construcción de herramientas teóricas que en lugar de usar el estado del mundo como su noción central formal, usen aspectos que el agente está sensando. (Brooks 1991a, 1991b).

Otro problema que se presenta en este nivel de diseño es la incertidumbre del mundo. En un principio la IABCn ignoró esta incertidumbre construyendo modelos del mundo "ingenierados". El principal problema que presenta la incertidumbre es la necesidad de construir un modelo del mundo. Esto significa que el sistema trata de tener una representación interna de una realidad objetiva externa. El problema se complica cuando las lecturas de los sensores presentaban problemas de ruido e introducían incertidumbre en la representación del mundo. Además, si se tenía más de un tipo de sensores, el problema se complicaba al tratar de combinar los diferentes tipos de datos en una única representación del mundo. Para librar estos obstáculos, es necesario la ausencia del modelo del mundo y operar en el mundo con una conexión directa a través de un ciclo de retroalimentación de percepción-acción.

Existen dos campos estrechamente relacionados a las arquitecturas de agentes del micro nivel: La teoría del control y la etología animal. La teoría del control puede ayudar a analizar la convergencia del comportamiento. La Etología estudia el comportamiento animal combinando las explicaciones causal y funcional, concentrándose principalmente en el comportamiento de animales en sus ambientes naturales.

2.4.3 Agentes individuales.

Después de haber provisto al agente artificial con comportamientos múltiples para manejar una variedad de circunstancias y lograr una variedad de tareas, en el macro nivel hay que decidir cuál comportamiento o comportamientos deben activarse en un momento determinado. Hay que considerar dos componentes en esta decisión: Cuáles comportamientos son potencialmente correctos dadas las circunstancias, y cómo resolver conflictos entre comportamientos.

Los desafíos en este nivel son:

a) Coherencia: El comportamiento del agente debe estar encaminado hacia la meta que se desea alcanzar. En este nivel en el diseño de la arquitectura del agente, es importante cuidar el hecho de no permitir que se activen dos comportamientos a la vez, de lo contrario habrá

interferencia ocasionando que ninguno de estos comportamientos opere con éxito⁷.

b) Preponderancia: Los comportamientos que están activos deben ser predominantes a la situación que el agente encuentra en sí mismo. En el caso de tratarse de un robot móvil, éste debe ser capaz de autorecargarse en caso de que sus baterías estén bajas y no cuando estén cargadas.

c) Suficiencia: El mecanismo de selección de comportamiento debe operar en tal forma que se consiga alcanzar las metas que le fueron instrumentadas al agente.

En este punto, la Etología también ayuda al tratar de explicar la selección de comportamientos por medio de una teoría que explica esta selección por medio de la motivación de un comportamiento en particular. Otra disciplina que aborda el mismo problema por medio de un modelo de control hormonal es la Neurociencia.

2.4.4 "Sociedades" de agentes.

Los sistemas sociales artificiales suelen inspirarse en las grandes colonias de insectos sociales, bandadas de pájaros, bancos de peces (Ünsal 1993) de ahí el término de "sociedades" de agentes. Este trabajo en particular se inspira en el comportamiento grupal de las colonias de hormigas.

En este nivel de diseño se trata la manera en que cada agente pueda interactuar con otro. Algunos puntos principales para conseguir tales interacciones son:

a) Emergencia: Dado un conjunto de comportamientos programados en un conjunto de agentes, debería ser posible predecir cuál será el

⁷ Un claro ejemplo de cómo incorporar estrategias para permitir la competencia y "desempate" de conductas que pueden activarse simultáneamente, se presenta en los mecanismos de selección de acción de Maes (Guerra 1997; Montes 1998). Tal mecanismo está constituido por una red distribuida, no jerárquica; cada conducta o módulo de competencia se representa por un nodo. Todas las conductas se encuentran al mismo nivel. Un módulo de competencia contiene un variable que define su nivel de activación, para que un módulo de competencia se active su variable de nivel de activación debe sobrepasar cierto umbral y ser mayor al de otros módulos que estén en condición de activarse. Cuando dos o más módulos de competencia tienen la misma posibilidad de activarse, se evita el conflicto eligiendo uno de ellos al azar. Además un módulo de activación inhibe a todos aquellos módulos con los que tiene conflictos.

comportamiento global del sistema. El desempeño global del sistema no se encuentra explícitamente programado sino más bien es resultado de las interacciones entre los individuos y de las interacciones que tienen estos con su medio ambiente.

b) Síntesis: El programa conducirá al conjunto de agentes a alcanzar una tarea determinada.

c) Individualismo: Es necesario determinar cuántas clases de agentes se necesitan para conseguir la realización de una tarea, considerando que los agentes de una clase en particular son idénticas entre sí.

d) Cooperación: Hay que proveer a los agentes de estrategias de cooperación para que logren obtener mejores resultados.

e) Interferencia: Los agentes pueden llegar a interferir entre ellos así que es necesario considerar protocolos de comunicación en su diseño.

f) Dependencia de densidad: El comportamiento global del sistema puede depender de la densidad de agentes y de las fuentes que consumen en el ambiente. Es probable que la tarea global no se logre por haber menos agentes que los necesarios, o en caso contrario, que no se logre porque los agentes se "estorban" para conseguirla.

g) Comunicación: El desempeño del sistema se puede incrementar al aumentar la cantidad de comunicación explícita⁸ entre agentes. Estos mecanismos de comunicación se analizan en la sección siguiente.

2.5 Enfoque Grupal de los agentes Inteligentes Basados en el Comportamiento.

Como se mencionó brevemente en el apartado 2.4.4, las "sociedades" de agentes toman como fuente de inspiración a las colonias o sociedades biológicas. Algunas sociedades de animales, particularmente los insectos sociales, pueden lograr tareas complejas que son imposibles de realizarse individualmente (Ünsal 1993). En el mundo biológico se observan abundantes casos de sistemas sociales que tienen pobres capacidades individuales comparadas con los comportamientos complejos que exhiben colectivamente. Esto se observa en diferentes estados de evolución, como en bacterias, en hormigas, en moluscos y en larvas; así como también en especies de nivel superior, como peces, pájaros y mamíferos (Colorni et. al. 1992). Ejemplos de las tareas complejas logradas por estos sistemas naturales son la recolección

⁸ La comunicación explícita se explica a detalle en la sección 2.5.2.

de alimento en las colonias de hormigas y de abejas, las formaciones logradas en los bancos de peces, los vuelos en las bandadas de pájaros, o las estructuras complejas realizadas por las termitas que son grandes edificadores de nidos (Ünsal 1993). Esto nos conduce a preguntarnos ¿Cuáles son las habilidades que los hace capaces de lograr una tarea en conjunto?. Deneubourg (Deneubourg et. al. 1992) analiza en particular la habilidad de edificar, la cual nos dice parece no estar relacionada con la capacidad cerebral, ya que los primates (a excepción de los humanos) o los delfines no son buenos edificadores, al contrario de organismos más simples como los artrópodos que son muy buenos edificadores. Es necesario entender cuáles son las reglas que gobiernan el comportamiento de los insectos sociales y que hacen que en conjunto sean capaces de edificar un nido o de recolectar alimento (Deneubourg et. al. 1992).

Lo anterior sugiere que podemos alcanzar metas similares de desempeño en tareas complejas, distribuyendo las actividades sobre sistemas masivamente paralelos, compuestos de elementos computacionalmente simples. Las sociedades naturales nos enseñan que un sistema con abundantes componentes simples, exhibirá un comportamiento global que se verá más **organizado** que el comportamiento de las partes individuales. Este fenómeno complejo es llamado comportamiento emergente del sistema (Colorni et. al. 1992). Diferentes procesos usados por insectos sociales se podrían usar para resolver diferentes problemas como la edificación o la producción de artefactos (Deneubourg et. al. 1992), la recolección de materiales peligrosos, la tarea de localización, la realización de exploraciones en misiones espaciales, la recolección de material, la búsqueda y la recolección de información - (Nilsson 1995) que es el caso de estudio que nos ocupa -, entre otros. Estos son diferentes ejemplos de inteligencia colectiva y descentralizada (Deneubourg et. al. 1992).

El enfoque grupal de los agentes inteligentes basados en el comportamiento se inspira en el mundo natural. Las sociedades de agentes para Maes (Maes 1993), son una clasificación de los sistemas multi-agente que persiguen una meta común, donde el comportamiento grupal inteligente emerge de sus interacciones, de manera que al perseguir reglas locales simples, los agentes pueden producir una solución emergente a tareas complejas (como encontrar un camino hacia una fuente de información, construir objetos, recolectar datos, entre otras. Nuestro trabajo utiliza la

metáfora del hormiguero para encontrar una solución colectiva al problema de la búsqueda y recolección de información en BD. Por ello, destacaremos a continuación las propiedades de las colonias de hormigas y enfatizaremos las ventajas de esta metáfora biológica para el caso que nos ocupa.

2.5.1 Colonias de hormigas.

Las colonias de hormigas son una de las sociedades animales más “organizadas” que conocemos. Las colonias de hormigas exhiben comportamientos muy interesantes, aún cuando una única hormiga tiene capacidades limitadas, el comportamiento global de la colonia es altamente estructurado: Es el resultado de interacciones coordinadas. Por otro lado, las posibilidades de comunicación entre las hormigas son muy limitadas, las interacciones deben basarse sobre flujos muy simples de información. En los sistemas inspirados en colonias de hormigas, cada individuo desempeña acciones muy simples y no conoce explícitamente lo que otros individuos están haciendo. Sin embargo, todo el sistema puede exhibir un comportamiento altamente estructurado (Colorni et. al. 1992).

El hormiguero se caracteriza por solucionar de manera colectiva un problema que, de manera individual no podría ser solucionado, al menos con la misma eficiencia. Para solucionar un problema por medio de un sistema artificial, es necesario considerar las reglas de comportamiento elementales que ayudarán al sistema a solucionar el problema en un ambiente determinado. La solución al mismo problema dependerá de si se trata de un agente individual o de una sociedad, así cómo también dependerá del tamaño de la sociedad (Deneubourg et. al. 1992).

Un “hormiguero artificial” se define como un conjunto de agentes, los cuales son capaces de comunicarse directa o indirectamente⁹. Lo anterior nos lleva a proponer una solución distribuida al problema. Las características esenciales de un “hormiguero artificial” son (Langton et. al. 1995):

- Es una colección de entidades relativamente autónomas no centralizadas que interactúan entre sí y con un ambiente dinámico.

⁹ La comunicación indirecta se explica a detalle en la sección 2.5.2.

Cada entidad se mueve aleatoriamente y no dispone de una representación global del ambiente, sólo conoce de éste lo que registran sus sensores (Deneubourg et. al. 1991).

- Comúnmente no hay una autoridad central dictando el comportamiento de la colección de individuos.

La organización descentralizada ofrece las ventajas de simplicidad, flexibilidad y robustez, a diferencia de una organización jerárquica donde la falla del coordinador central implica la falla de todo el sistema (Deneubourg et. al. 1991; Ünsal 1993). En estos sistemas, ningún componente es el encargado de lograr el comportamiento complejo (Maes 1993).

- Aunque los individuos son relativamente simples, su comportamiento colectivo puede ser complejo.
- La combinación de los comportamientos individuales determina el comportamiento colectivo del grupo en conjunto.
- El diseño del sistema está encaminado a permitir una operación eficiente en máquinas de arquitecturas paralelas. Una característica importante del hormiguero artificial consiste en que los agentes que lo integran actúan paralelamente sobre los elementos de su ambiente (Deneubourg et. al. 1992). Los individuos trabajan en paralelo y sin obedecer a una organización jerárquica (Deneubourg et. al. 1991).

En el hormiguero artificial, los individuos o agentes son capaces de auto-organizarse. La auto-organización es una característica importante en algunas sociedades animales, como las antes citadas, así como en la mayoría de las sociedades de insectos. Las tres características importantes que presenta un sistema auto-organizable son (Ünsal 1993):

- Un sistema auto-organizable puede realizar tareas a partir de comportamientos individuales mínimos y simples.
- Un cambio en el ambiente puede influir en el sistema para que genere una tarea diferente, sin cambiar las características del comportamiento.
- Cualquier diferencia, por pequeña que sea, puede influir en el comportamiento colectivo del sistema.

En estos sistemas dinámicos, pequeños cambios en el nivel individual pueden modificar en gran medida el comportamiento global de la población (Drogoul & Ferber 1993).

Sin embargo, debemos señalar que en un sistema auto-organizable el comportamiento individual no necesita cambiarse para obtener un comportamiento colectivo diferente. Esta característica lo hace altamente ventajoso, al permitir el logro de un comportamiento individual simple con diseños relativamente baratos. Un gran número de individuos se pueden coordinar en un sistema colectivo que interactúa con su ambiente. El comportamiento colectivo que se obtiene tiene un carácter adaptativo, confiable y simple que sólo necesita de unas cuantas reglas básicas para definir el comportamiento de cada individuo y las interacciones con el ambiente. La auto-organización alcanzada por el hormiguero artificial muestra que las unidades descentralizadas, con interacciones directas con el ambiente (retroalimentación), son capaces de producir patrones complejos y solucionar problemas (Deneubourg et. al. 1991). **El hormiguero representa la idea de un orden emergiendo de un caos** (Hofstadter 1989).

Un hormiguero artificial alcanza la auto-organización funcional o la inteligencia distribuida, esto es, la organización del hormiguero artificial emerge de su dinámica interna y de su interacción con el ambiente. (Deneubourg et. al. 1991, 1992).

Los resultados en el nivel grupal son más substanciales. Esto implica que se debe ser muy cuidadoso al diseñar un hormiguero artificial o cualquier sistema distribuido en el cual la inteligencia es colectivamente exhibida por entidades no inteligentes. Es necesario también, examinar el cómo diferentes tipos de restricciones operan sobre el comportamiento individual y sobre la dinámica del grupo (Drogoul & Ferber 1993).

En la investigación inspirada en la metáfora del hormiguero, los agentes exploradores y recolectores de información parecen ser uno de los ejemplos más comunes para ilustrar la capacidad de una población de agentes de pobre inteligencia para conseguir una meta global (Drogoul & Ferber 1993), y es precisamente en este sentido en el que utilizamos a nuestros agentes para solucionar el problema que nos hemos planteado.

2.5.2 Métodos de comunicación entre agentes.

Abordaremos a continuación los mecanismos de comunicación empleados por las sociedades de insectos sociales, pues el estudio es útil en el diseño de nuestros agentes recolectores.

Las hormigas pueden explorar de manera aleatoria, existen sin embargo tendencias globales que pueden emerger de ese caos y abarcan a grandes cantidades de ellas. Un ejemplo de esto son los senderos que forman, los movimientos de cada hormiga en particular son del todo imprevisibles, pero el sendero mismo da la impresión de permanecer bien definido y estable. Indudablemente, eso implica pensar que las hormigas no se mueven en forma del todo aleatoria. Es decir, que existe cierto grado de comunicación entre las hormigas, el cual es necesario precisamente para evitar que exploren caprichosamente (Hofstadter 1989). El grado mínimo de comunicación entre las hormigas lo obtienen de la estrategia de reclutamiento (Muñoz 1997; Wilson 1998a) descrita en la sección 1.2.3.

La “inteligencia” colectiva de un hormiguero es un comportamiento emergente de la comunicación colectiva (Ünsal 1993). Es importante descubrir los mecanismos de comunicación que debemos implantar en las colonias de hormigas artificiales que permita la interacción entre los agentes-hormiga, que los encamine a lograr una auto-organización y el cumplimiento de una meta fijada. Además la comunicación es el medio más común de la interacción entre agentes inteligentes.

Los mecanismos de comunicación que se implementaron en nuestros agentes-hormiga, se inspiraron en la estrategia de reclutamiento mencionada. Con el propósito de ubicar el tipo de comunicación que se empleó, citaremos la clasificación que refiere Mataric sobre métodos de comunicación (Mataric 1994, 1995):

- a) *Comunicación directa*: es un acto puramente comunicativo, con el único propósito de transmitir información, tal como el acto de hablar o la transmisión de un mensaje de radio. Este tipo de comunicación se dirige hacia un receptor en particular. Puede ser **uno a uno, o uno a muchos**, pero en ambos casos los receptores son identificados.

- b) En contraste la *comunicación indirecta* se basa en el comportamiento observable (no en la comunicación) de otros agentes, y en sus efectos sobre el ambiente. Este tipo de comunicación se funda en las modificaciones al ambiente en lugar del envío de un mensaje directo.

La **cooperación** es una forma de interacción, usualmente basada en la comunicación. Análogamente a la comunicación, Mataric (Mataric 1995) expone dos tipos de cooperación:

- a) Una *cooperación explícita* se define como un conjunto de interacciones, las cuales involucran intercambio de información o de acciones que se desempeñan para beneficiar a otros agentes.
- b) Una *cooperación implícita* consiste de acciones que son una parte del repertorio de comportamientos del propio agente, pero que tiene efectos en el ambiente que ayudan a otros agentes a conseguir sus metas.

Los agentes-hormigas que se presentan en este trabajo utilizan la comunicación y la cooperación implícitas. La cooperación también ha sido abordada como una funcionalidad emergente de un sistema (Muñoz 1997), es decir como un comportamiento no explícitamente programado sino como resultado de las interacciones locales entre los componentes del sistema. La funcionalidad del sistema es entendida como el rango de tareas que el agente puede realizar (Para abundar al respecto ver Steels 1994).

La idea básica en la comunicación implícita consiste en que las interacciones indirectas son necesarias para coordinar el trabajo del grupo. Los agentes están gobernados por sus percepciones e interacciones locales (Drogoul & Ferber 1993). Con una sucesión de estímulos - reacciones, la sociedad debe ser capaz de producir una estructura. El resultado se logra por la coordinación de las actividades individuales de la sociedad. Las sociedades biológicas muestran mecanismos simples que son capaces de solucionar problemas y que son mucho más poderosos de lo que pensamos. Se intenta reproducir las reglas de comportamiento que produzcan estructuras similares. (Deneubourg et. al. 1992; De la Mora 1997).

Las hormigas usan caminos de feromona para comunicar información entre los individuos. Mientras una hormiga se mueve esencialmente de forma

aleatoria, puede detectar una marca leve de feromona y decidir con una alta probabilidad seguirla, y reforzarla con su propia feromona. La probabilidad con la cual una hormiga elige una marca aumenta con el número de hormigas que eligen el mismo camino en pasos anteriores. La decisión de seguir una marca no es determinística, permitiendo una continua exploración de rutas alternativas. Esto es un resultado importante, donde un nivel mínimo de complejidad individual puede explicar un comportamiento colectivo complejo. La metáfora del hormiguero plantea una solución distribuida a un problema difícil con la operación de muchos agentes simples interactuando localmente (Coloni et. al. 1992).

Nuestros agentes se comunican de manera indirecta, a través de su efecto sobre el ambiente. La simplicidad del diseño de cada agente-hormiga hace que sean baratos y robustos, y al no estar controlados jerárquicamente, previenen que el sistema completo se paralice al fallar alguna unidad.

En conjunto se tiene un sistema simple, flexible, tolerante a errores y una altamente rentable. Esta es una característica general de los sistemas en los cuales el comportamiento colectivo de un grupo de agentes autónomos emerge en lugar de estar explícitamente programado, *y es seguramente una de las razones por la que 100 millones de años de evolución dieron el triunfo a los insectos sociales* (Deneubourg et. al. 1991).

2.5.3 Resultados Sinérgicos.

El resultado Sinérgico se da desde un punto de vista holístico (Hofstadter 1989), donde “El todo es mayor que la suma de sus partes”.

Las colonias de agentes artificiales están formadas por individuos que sólo son capaces de desempeñar acciones muy simples, desconocen lo que otros agentes hacen y tienen pobres capacidades cognitivas. Sin embargo, de la operación en conjunto puede observarse un resultado altamente estructurado. La sinergia obtenida por las colonias de agentes se observa en el resultado del cumplimiento de una tarea, donde dicha tarea se termina de mejor manera precisamente en conjunto, con mecanismos de comunicación y cooperación entre ellos.

Es importante determinar la manera en que influye el tamaño de la colonia al alcanzar una tarea, en comparación con el desempeño de un individuo para conseguir la misma tarea. En (Deneubourg et. al. 1992) se analiza un experimento sobre una colonia de termitas, cuyos individuos trabajan conjuntamente para conseguir edificar un nido. Los resultados muestran que el poder de un individuo del grupo es mayor que el poder de un solo individuo. Esto se consigue cuando los individuos unen sus esfuerzos, es decir, el mismo individuo consigue más al trabajar en grupo que individualmente.

Capítulo 3.

Una “Sociedad” de Agentes Inteligentes Recolectores de Información.

3.1 Descripción del problema.

Se ha mencionado en la motivación del trabajo (sección 1.1), que el presente trabajo tiene como objetivo el diseño de agentes capaces de desempeñarse en ambientes imprevisibles y complejos. Tales agentes utilizan estrategias de bajo costo de diseño y de bajo costo computacional. Para poder experimentar con ello, hemos implementado un sistema multi-agente Basado en el Comportamiento que está orientado a resolver la tarea de búsqueda y recuperación de información en una BD.

El sistema utiliza la metáfora del hormiguero y como se mencionó en la sección 1.2.3 esta inspiración biológica es oportuna al permitirnos instrumentar sistemas de bajo costo, de pobres capacidades cognoscitivas y eficientes para resolver colectivamente el problema.

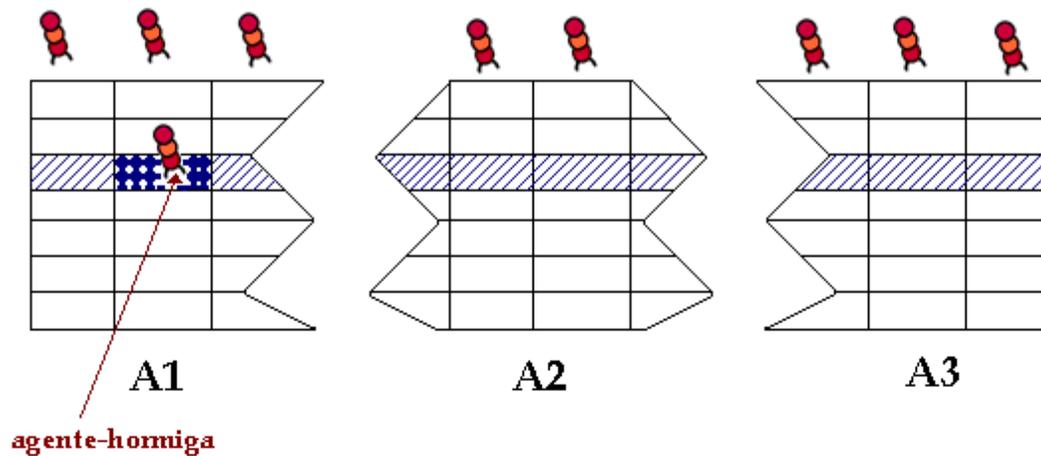
Se tiene una BD distribuida localmente, esta BD contiene información no adecuada para ser recuperada por los métodos tradicionales de recuperación de información, ésta información puede estar oculta entre otros datos, fragmentada, escrita en diferente forma (con letras en mayúsculas, minúsculas o intercalando ambas), entre otros casos.

Para recuperar la información se cuenta con una colonia homogénea de agentes-hormiga, estos agentes-hormiga “saben” recolectar la información que coincida con el dato que indica el usuario. Además los agentes-hormiga

pueden marcar información que consideran relevante para comunicar a otros agentes-hormiga de la utilidad de la información. Con este mecanismo indirecto de comunicación se espera que emerja un comportamiento de cooperación entre los agentes, un comportamiento grupal “inteligente” a partir de agentes con mínimas capacidades cognoscitivas.

Además se plantea que los agente-hormiga realicen una exploración exhaustiva sobre la BD distribuida con el propósito de aprovecharla al máximo. Es cierto que no todos los agente-hormiga encontrarán la información que se busca como tal, pero si contribuirán en su recolección y con ello cada agente-hormiga contribuye al cumplimiento de la tarea global. Cada agente-hormiga encuentra sólo una porción de información y entre todos los agente-hormiga construyen el resultado de la búsqueda mostrando la solución al usuario. El esquema del problema se muestra en la figura 3.1.

BD Fragmentada.



Resultados.

[Red hatched pattern]						

Información recolectada.

A1, A2 y A3 : Partes de la BD.

-  Información relevante encontrada por el agente-hormiga.
-  Información marcada por el hallazgo del agente recolector en A1
-  Información recolectada por el agente que explora en A1, además de la información asociada al hallazgo del agente en A1

Figura. 3.1. Esquema del problema a solucionar con la ayuda de los agentes-hormiga.

3.1.1. Propuesta de solución.

Para solucionar el problema de búsqueda y recuperación de información proponemos un sistema multi-agente que utiliza la metáfora del hormiguero. Este sistema ha sido creado utilizando la metodología de la IABCm, ha sido además comprendido como un instrumento para estudiar aspectos relativos al comportamiento grupal en sistemas basados en el comportamiento y mostrar así que estos sistemas son dignos de considerarse en la solución de problemas complejos. Las características que presenta nuestra “sociedad” de agentes recolectores son:

- Cada agente que forma parte de la “sociedad” no tiene conocimiento global del “mundo” o ambiente en el que habita, sólo conoce de él lo que puede percibir mediante sus sensores.
- No hay controladores centrales, ni jefes, ni castas en nuestra sociedad. Se tiene una colonia homogénea¹⁰ de agentes-hormiga.
- La solución al problema complejo se consigue a través de la colaboración de cada individuo de la colonia.
- La colaboración se alcanza con la implementación de mecanismos de comunicación mínimos. La colaboración que se propone es una colaboración implícita.

Es importante mencionar que nuestra colonia se beneficia del desempeño de un agente individual basado en el comportamiento, este agente es el encargado de “preparar el camino”. Esto quiere decir que dicho agente individual buscará sinónimos definidos previamente en una base de datos y los cargará en memoria para ser aprovechados por la colonia y generar resultados más versátiles a las búsquedas. Este agente será nombrado en adelante **agente buscador de sinónimos**.

3.1.2. Metas de la instrumentación.

Nuestra colonia de individuos recolectores ha sido planteada como un instrumento para el estudio de los siguientes aspectos:

¹⁰ La homogeneidad en la colonia se refiere al hecho de que los individuos que la conforman son semejantes en los perceptores, actuadores y conductas de los que han sido dotados.

- Estudiar el cómo a partir de un diseño simple se puede solucionar un problema complejo.
- Instrumentar mecanismos de comunicación indirecta para conseguir la interrelación entre los individuos y el ambiente en el que se desempeñan.
- Estudiar el cómo, a partir de mecanismos de comunicación muy simples, puede emerger el fenómeno de cooperación entre los individuos.

3.1.3 Diseño.

Para el diseño de nuestra colonia de agentes-hormiga, hemos seguido la agenda de trabajo de Brooks (Brooks 1991b) la cual fue presentada en la sección 2.4. Esta agenda de trabajo se resume en seguida:

- Primero se deben diseñar e implementar los componentes necesarios del agente -perceptores y efectores- estableciendo sus conductas básicas.
- En segundo término debe diseñarse un mecanismo de selección de conductas para permitirle al agente conducirse de manera adecuada en cada momento.
- Finalmente, deben instrumentarse mecanismos de comunicación para que un agente pueda agruparse con otros agentes idénticos o similares. En este nivel se estudia la forma de comunicación entre los individuos para hacer emerger un comportamiento grupal complejo, se diseñan "sociedades" de agentes que exhiban comportamientos colectivos y de cooperación.

A continuación se describe el trabajo desarrollado en cada punto de la agenda.

3.1.3.1. Componentes del agente.

En esta parte del diseño es necesario establecer los mecanismos mínimos que le permitan al agente interactuar con su ambiente. Con tales mecanismos el agente debe ser capaz de percibir su entorno y de actuar en él, debe ser capaz también de desempeñarse en ese mundo a pesar de ser complejo. Como parte del diseño del agente se ha considerado reducir su cognoscitividad al mínimo, es decir, cada agente declinará el uso de un mapa o representación del ambiente en el que se desempeñará. Recordemos que debido a que los sistemas basados en el comportamiento se caracterizan por estar situados en su medio ambiente, no requieren de un modelo del mundo, “el mundo es su mejor modelo” (Brooks 1991a).

En esta parte se diseñan los requerimientos mínimos perceptuales y conductuales de un agente-hormiga. Es necesario pues, dotar al agente de los receptores y actuadores esenciales para poder reaccionar a los cambios en el ambiente, permitiendo la interacción con ambientes dinámicos e imprevisibles.

El diseño de los componentes esenciales de nuestro individuo se basó en la heurística de diseño de mundos incrementalmente complejos de De la Mora (De la Mora 1997). La idea básica de la heurística consiste en establecer una definición clara del tipo de tareas que se desea realice el individuo, así como el medio ambiente en el que contendrá. El proceso se inicia considerando el medio ambiente “más sencillo” posible y el espacio de percepciones requerido para obtener un primer esbozo del agente que satisfaga las metas que el diseñador tiene. A partir de esto el medio ambiente se complica incrementalmente, con el consiguiente aumento del espacio de percepciones, de manera que en cada ciclo se define de mejor manera al agente o individuo.

El enfoque de De la Mora (De la Mora 1997) considera a la inteligencia como un atributo que puede darse en forma gradual en diferentes seres vivos y no como una propiedad “exclusiva” de los seres humanos, asociándola con la supervivencia o funcionalidad que presente el ser vivo. El diseño de cada agente inicia con un conjunto de reglas reactivas con las que se implementa un conjunto de comportamientos que le permitan la operatividad mínima o “supervivencia” del sistema en su medio. Después se introducen nuevas

capacidades más complejas añadiendo módulos de tareas específicas los cuales, cuando se combinan, resultan en el comportamiento global deseado. Se incrementa el diseño del agente al incrementar la complejidad del mundo consiguiendo el refinamiento del propio diseño.

La descripción de los requerimientos y capacidades de cada agente hormiga aplicando la heurística de definición de mundos incrementalmente complejos, es presentada en las tablas 3.2 y 3.3. La tabla 3.3 muestra las conductas poseídas por nuestros individuos. El término conducta denota secuencias de acciones espacio-temporalmente coordinadas. Una conducta puede estar conformada por la combinación de conductas básicas o de otras conductas. Por conductas básicas entenderemos las acciones que pueden ser llevadas a cabo por nuestros individuos activando sus efectores o a través de la activación combinada de sus efectores. (Muñoz 1997).

En adelante, se manejarán los siguientes términos:

- **Mundo:** Ambiente o BD donde el agente explorará.
- **Información:** Se refiere a la información que el agente debe buscar y recolectar en la mundo.
- **Feromona:** Marcas que los individuos dejarán en el mundo mientras recolectan información y que los guiará hacia la fuente de ésta.
- **Dato Preciso:** Es la información que se encuentra en el mundo y que define de modo exacto al dato que se está buscando.
- **Dato Equivalente:** Es la información que se encuentra en el mundo y que es afín al dato que se busca. La información en el mundo puede contener letras en mayúsculas, minúsculas e incluso combinar mayúsculas con minúsculas, y diferir en ello con el dato a buscar.
- **Dato Similar:** Es la información que se encuentra en el mundo y que se relaciona con el dato a buscar. La información que se encuentra en el mundo contiene los mismos caracteres (y en el mismo orden) que el dato a buscar, pero separados por espacios en blanco o tabulaciones.
- **Dato Oculto:** Es la información semejante al dato a buscar y que se encuentra en el mundo contenida dentro de alguna frase o entre otros datos.

- **Dato Sinónimo Preciso:** Es la información que se encuentra en el mundo de igual significado que el dato a buscar. Este sinónimo es exactamente igual al sinónimo encontrado por el agente buscador de sinónimos.
- **Dato Sinónimo Equivalente** Es la información que se encuentra en el mundo de igual significado que el dato a buscar. La información en el mundo puede contener letras en mayúsculas, minúsculas e incluso combinar mayúsculas con minúsculas, y diferir en ello con el sinónimo encontrado por el agente buscador de sinónimos..
- **Dato Sinónimo Similar** Es la información que se encuentra en el mundo y que se relaciona con el dato a buscar. Este sinónimo contiene los mismos caracteres (y en el mismo orden) que el dato a buscar, pero separados por espacios en blanco o tabulaciones y es análogo al sinónimo encontrado por el agente buscador de sinónimos.
- **Dato Sinónimo Oculto** Es la información semejante al sinónimo encontrado por el agente buscador de sinónimos y que se encuentra en el mundo contenida dentro de alguna frase o entre otros datos.

En la tabla 3.1 se muestran algunos ejemplos de las definiciones anteriores. Supóngase que se tiene lo siguiente:

Mundo : Mibase.dbf
 Dato a buscar : **alto**.

Mundo = Mibase.dbf				
Nombre	Dirección	Colonia	Estatura	C.P.
Daniel	Gardenia # 3	Los altos ⁴	alto ¹	69000
Carlos	Juárez # 2	Los superiores ⁸	a l t o ³	69877
Juan	Porfirio Díaz #1	Centro	AlTo ²	65000
Alberto	Gardenia # 23	SuPeriores ⁶	superior ⁵	69000
Enrique	Juárez # 12	s u p e r i o r e s ⁷	bajo	69000

Tabla 3.1. Ejemplos de la terminología utilizada. La numeración en superíndices corresponde a: 1= Dato Preciso, 2=Dato Equivalente, 3=Dato Similar, 4=Dato Oculto, 5=Dato Sinónimo Preciso, 6=Dato Sinónimo Equivalente, 7=Dato Sinónimo Similar, 8=Dato Sinónimo Oculto.

Mundo	Descripción	Requerimientos
0	Mundo	Sensor de mundo, mecanismos para moverse a través del mundo.
1	Mundo con Dato Preciso	Requerimientos del mundo 0, sensor de información precisa, mecanismos para recolectar la información, mecanismos para marcar con feromona, mecanismos para indicar que la información fue recolectada.
2	Mundo con Dato Equivalente	Requerimientos del mundo 1, sensor de información equivalente.
3	Mundo con Dato Similar	Requerimientos del mundo 2, sensor de información similar.
4	Mundo con Dato Oculto	Requerimientos del mundo 3, sensor de información oculta.
5	Mundo con Dato Sinónimo Preciso.	Requerimientos del mundo 4, sensor de sinónimo preciso.
6	Mundo con Dato Sinónimo Equivalente.	Requerimientos del mundo 5, sensor de sinónimo equivalente.
7	Mundo con Dato Sinónimo Similar.	Requerimientos del mundo 6, sensor de sinónimo similar.
8	Mundo con Dato Sinónimo Oculto.	Requerimientos del mundo 7, sensor de sinónimo oculto.
9	Mundo con feromona.	Requerimientos del mundo 8, sensor de información marcada con feromona.
10	Mundo Con información relevante agotada.	Requerimientos del mundo 9. Sensor de información agotada. Mecanismos para detener la exploración de los agentes en la BD.

Tabla 3.2. Requerimientos de nuestros individuos, identificados a través de la heurística de definición de mundos incrementalmente complejos de De la Mora (De la Mora 1997).

Mundo	Conductas básicas.
Mundo 0	AvanzarRegistro, AvanzarCampo.
Mundo 1	Conductas del mundo 0, RecolectarDatoPreciso, MarcarDatoPreciso, MarcarInformacionRecolectada.
Mundo 2	Conductas del mundo 1, RecolectarDatoEquivalente, MarcarDatoEquivalente.
Mundo 3	Conductas del mundo 2, RecolectarDatoSimilar, MarcarDatoSimilar.
Mundo 4	Conductas del mundo 3, RecolectarDatoOculto, MarcarDatoOculto.
Mundo 5	Conductas del mundo 4, RecolectarDatoSinónimoPreciso, MarcarDatoSinónimoPreciso.
Mundo 6	Conductas del mundo 5, RecolectarDatoSinónimoEquivalente, MarcarDatoSinónimoEquivalente.
Mundo 7	Conductas del mundo 6, RecolectarDatoSinónimoSimilar, MarcarDatoSinónimoSimilar.
Mundo 8	Conductas del mundo 7, RecolectarDatoSinónimoOculto, MarcarDatoSinónimoOculto.
Mundo 9	Conductas del mundo 8, RecolectarDatoConFeromona.
Mundo 10	Conductas del mundo 9, DetenerExploración.

Tabla 3.3. Definición de conductas básicas en nuestros individuos, identificados a través de la heurística de definición de mundos incrementalmente complejos de De la Mora (De la Mora 1997).

3.1.3.2. Agentes Individuales.

Las conductas de los individuos son organizadas a través de un mecanismo de selección y activación de conductas. Es necesario por tanto, dotar al agente de un mecanismo para la selección de comportamientos, la que hemos usado en nuestro sistema es llamada Arquitectura Subsumción (Brooks 1986), que tiene un esquema de prioridad fijo para “arbitrar” entre comportamientos.

En esta arquitectura se construyen capas de control del sistema para que el agente opere en un espectro de habilidades. Un espectro de habilidades es un especificación informal de una clase deseada de comportamientos para un agente que se encuentre en cualquier ambiente. Un alto espectro de habilidades implica una clase de comportamientos deseados más específicos. La idea es incrementar el espectro de habilidades del individuo al especificar capas de conductas que éste podrá desempeñar.

La arquitectura subsumción es un método para construir sistemas de control de robots móviles o agentes, vinculando las actividades de percepción y acción. Tal sistema de control está formado por capas de módulos asíncronos (mediadores) que operan incrementando su espectro de habilidades : Un espectro de habilidades es una especificación informal de una clase deseada de conductas para un agente que enfrentará diversos ambientes (Muñoz 1995).

Cada mediador es una instancia de una máquina computacional relativamente simple. La organización de los mediadores en capas responde a la especificación de una jerarquía de conductas: El tope de la jerarquía representa en términos generales el objetivo que el agente habrá de cumplir, el cuál será alcanzado con la ayuda del conjunto de mediadores ubicados en niveles inferiores. Cada capa agregada representa una clase de conducta más específica. Se dice que los mediadores de menor jerarquía están subsumidos o subordinados a los de mayor nivel. Las capas de los niveles superiores subsumen los papeles de las capas inferiores inhibiendo sus salidas; no obstante, éstas últimas continúan en funcionamiento. Lo anterior se esquematiza en la figura 3.2 (Brooks 1986).

Cada nivel de la arquitectura subsumción genera un comportamiento y la competencia del agente está dada agregando nuevas capas. Esta arquitectura se basa en la descomposición del agente en términos de comportamiento. El control de todo el sistema puede verse como un sistema de agentes actuando separadamente, no necesita de un módulo de control central, pues cada módulo actúa de manera descentralizada (Ünsal 1993).

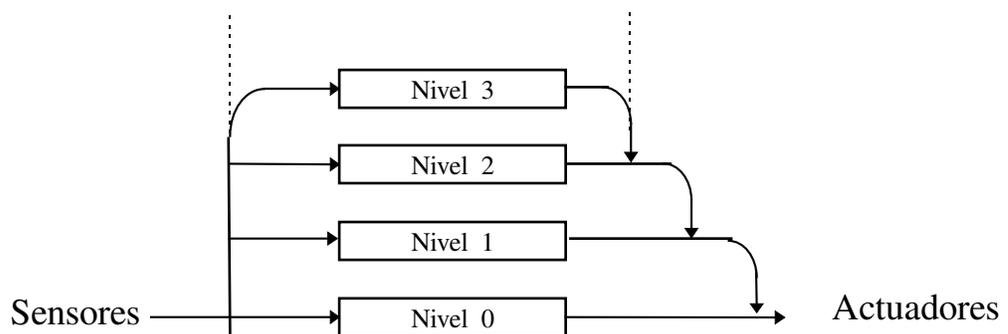


Figura 3.2. Esquema que muestra la estructura de la arquitectura subsumción. Las capas de conductas superiores, cuando toman el control, subsumen los papeles de los niveles inferiores (Brooks 1986).

El orden de evaluación de las conductas de uno de nuestros agente-hormiga identificadas y nombradas en la sección anterior, está organizado de acuerdo a la siguiente jerarquía.

1. RecolectaDatoPreciso
2. RecolectaDatoEquivalente
3. RecolectaDatoSimilar
4. RecolectaDatoOculto
5. RecolectaDatoSinonimoPreciso
6. RecolectaDatoSinonimoEquivalente
7. RecolectaDatoSinonimoSimilar
8. RecolectaDatoSinonimoOculto
9. RecolectaDatoConFeromona
10. DetenerExploracion
11. AvanzarCampo
12. AvanzarRegistro

3.1.3.3 “Sociedad” de agentes hormiga.

Nuestro sistema emplea la técnica de reclutamiento inspirada en la que usan los insectos sociales y que se describió en la sección 1.2.3., para conseguir la colaboración entre los agentes al recolectar información se utiliza la comunicación indirecta.

En nuestro estudio el ambiente sobre el cual la colonia realizará sus búsquedas en una base de datos distribuida localmente. El objetivo de la colaboración de los agente-hormiga se da no sólo al buscar la información, sino también al recolectarla, esto es, entre todos los agentes deben ser capaces de reunir la información de los diferentes segmentos de la BD.

La manera cómo se consigue la colaboración entre los agentes-hormiga se muestra en la figura 3.3., en dicha gráfica podemos apreciar que cuando un agente-hormiga encuentra en un determinado registro un dato con información relevante (Dato Preciso, Dato Equivalente, Sinónimo Preciso, ...), marca los otros campos de ese registro cómo información asociada, de modo que cuando otro agente-hormiga u otros agentes-hormigas se encuentren explorando en un campo que pertenezca a dicho registro perciban

que es información que aunque no es la que se busca, está asociada con ella y debe ser presentada al usuario. De esta manera se construye el resultado de la búsqueda, un agente-hormiga recupera y marca la información que encontró como relevante y el resto de nuestra colonia artificial colaboran en la recuperación de la información asociada a dicho hallazgo.

3.1.3.4. Agente buscador de sinónimos.

Antes se habló brevemente sobre este agente que “prepara el terreno” para la búsqueda de nuestra colonia de agentes-hormiga. Este agente buscador de sinónimos es también un agente basado en el comportamiento. Sus sensores son semejantes a un agente-hormiga y es capaz de sensor un Dato Preciso, un Dato Equivalente, un Dato Similar o un Dato Oculto. Este agente busca en una BD especial que se ha diseñado para este propósito. Esta BD está en formato dBase, la información que contiene está ordenada alfabéticamente (con el fin de facilitar la búsqueda al agente), cada registro inicia con una palabra seguida de sus sinónimos los cuales están separados por comas.

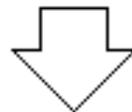
Al terminar de explorar la BD de los sinónimos, el agente buscador de sinónimos creará una lista de los sinónimos que corresponden al dato que el usuario desea buscar. Los agentes-hormiga puede tener acceso a esta lista y con ello pueden realizar búsquedas más versátiles. Sin embargo, si el agente buscador de sinónimos no encontró sinónimos para el dato que desea buscar, no se verá afectado el comportamiento de los agentes-hormiga.

Para el diseño de este agente, también se siguió la heurística de diseño de mundos incrementalmente complejos de De la Mora (De la Mora 1997), descrita en la sección 3.1.3.1, para identificar las conductas básicas requeridas. El mecanismo activador de conductas que se empleó en el agente buscador de sinónimos también es la arquitectura subsumción descrita en la sección 3.1.3.2.

Colaboración de los agentes-hormiga



Nombre	Dirección	Colonia	Estatura	C.P.
Daniel	Gardenia # 3	alto ¹	Los altos	69000
Juan	Porfirio Díaz #1	Centro	Alto	65000
Alberto	Gardenia # 23	SuPeriores	superior ²	69000
.
.



Información Recolectada.

Nombre	Dirección	Colonia	Estatura	C.P.
Daniel	Gardenia # 3	alto ¹	Los altos	69000
Alberto	Gardenia # 23	SuPeriores	superior ²	69000

Dato a buscar : "alto"

-  Información relevante encontrada en la BD y que es igual al dato a buscar.
-  Información relevante encontrada en la BD y que es un sinónimo del dato a buscar.
-  Información marcada con feromona, y que se considera información asociada.

1 = Dato Preciso.

2 = Dato Sinónimo Preciso

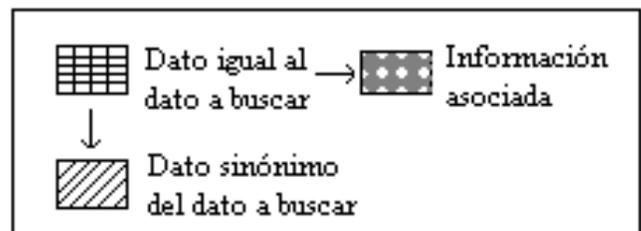


Figura 3.3. Formación de "caminos" de feromona sobre la BD.

3.1.4 Implementación.

La implementación de nuestra colonia de agentes-hormiga fue realizada en Delphi Ver. 3.0. Este ambiente nos permite crear aplicaciones sofisticadas bajo un ambiente de diseño visual. Delphi integra una herramienta llamada *Borland Database Engine*, que nos permite trabajar con BD en formato dBase, Paradox, y también con BD locales ODBC. También se incluye en ésta versión de Delphi a *Local InterBase Server* con el que pueden crearse aplicaciones cliente/servidor. Otra herramienta importante que se incluye es *ReportSmith* el cual permite a los programadores preparar reportes para cualquier formato de BD. Estas herramientas adicionales pueden ser consideradas en futuras extensiones de nuestro sistema.

El lenguaje de programación que maneja Delphi es Pascal Orientado a Objetos. El hecho de que Delphi permitiera la programación orientada a objetos (POO) fue beneficioso ya que la POO permite una definición natural del tipo de sistemas que aquí se estudian (Para una discusión sobre la conveniencia de modelar sistemas mediante técnicas de POO véase Muñoz 1995).

Un objeto en la POO posee características y exhibe un comportamiento definido, una clase es una descripción de un conjunto de objetos que comparten propiedades y conductas comunes. Así pues, la definición de la clase agente permite la creación de n agentes-hormiga semejantes, lo cual deseamos para la creación de nuestra colonia artificial.

Fueron dos las clases que se definieron, la clase agente que define a cada uno de nuestros agentes-hormiga y la clase agente-busca-sinonimos que define al agente que prepara el terreno para la búsqueda. La clase agente-busca-sinonimos es una subclase de la clase agente. La definición de estas clases se muestra en el Apéndice A.

Por otra parte, para elegir la plataforma en la que se realizaría el sistema fue necesario hacer algunas pruebas, mismas que se explican a detalle enseguida.

3.1.4.1 Elección de plataforma de trabajo.

Con el propósito de elegir la plataforma en la cual se llevó a cabo la codificación del sistema se realizaron tres pruebas:

1. La programación de la interfaz codificada en C++ para DOS.
2. La programación de la interfaz codificada en Tk/Tcl (Shareware) para Windows.
3. La programación de la interfaz codificada en Delphi 3.0 para Windows.

La interfaz de prueba consistía de una ventana como la mostrada en la figura 3.4., está interfaz consistió de un menú con las opciones siguientes:

- Abrir_BD (Abrir base de datos)
- Buscar
- Salir

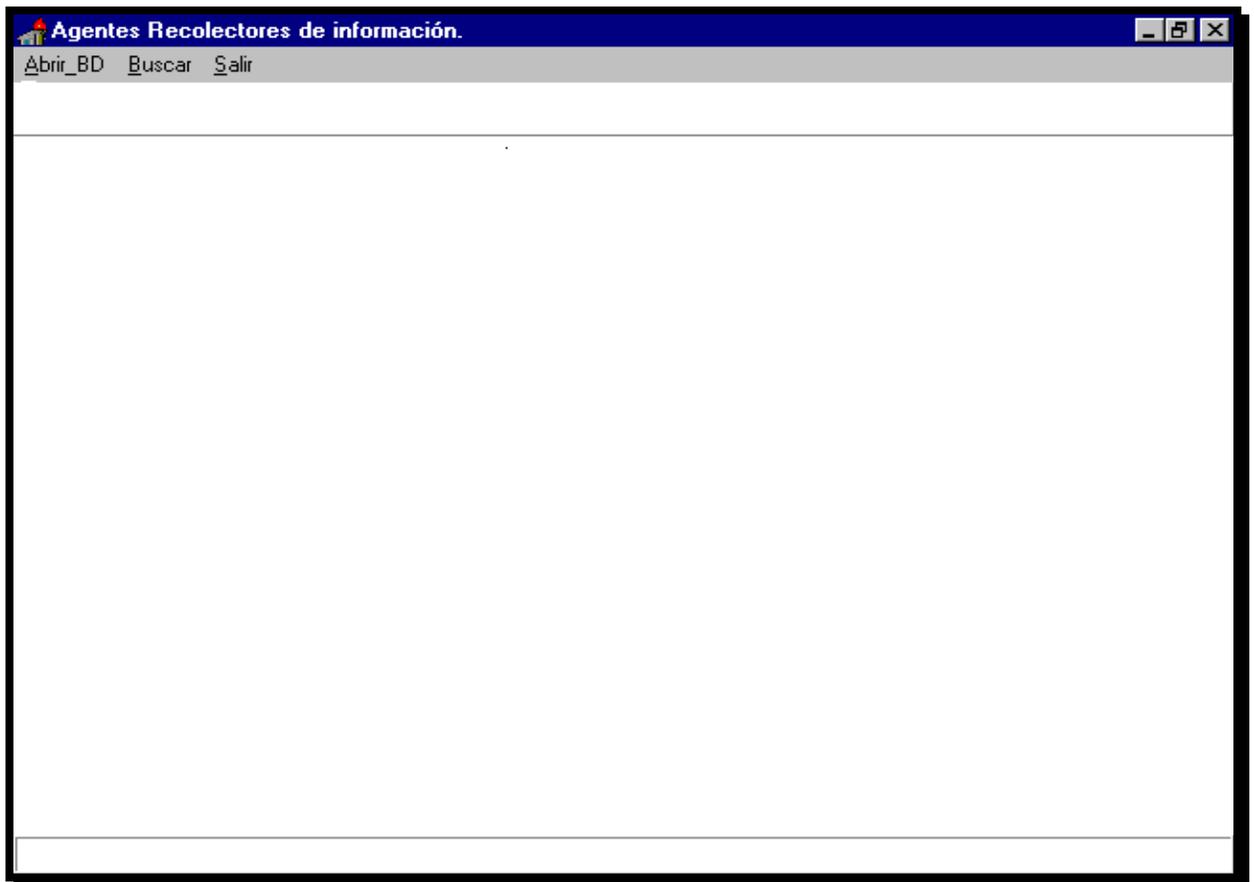


Figura 3.4. Modelo de la interfaz de prueba creada bajo C++ para DOS, Tk/Tcl y Delphi

Para tomar una decisión sobre la plataforma a elegir se consideraron los siguientes criterios de evaluación.

- a) Espacio en disco y de memoria requeridos de la instalación del paquete de programación.
- b) Portabilidad (Si el sistema puede trabajar en DOS, WINDOWS, UNIX ...).
- c) Desarrollo de la aplicación (Tiempo y facilidad de programación).
- d) Interacción de la interfaz con el usuario (Qué tan amigable le resulta al usuario).
- e) Dominio del lenguaje de programación.
- f) Disponibilidad (Discos de instalación y manuales).
- g) Lenguaje de programación depurable.

La tabla 3.4 muestra las características de cada ambiente de desarrollo en el que se realizó la interfaz prueba, y la tabla 3.5 muestra las puntuaciones que obtuvo cada uno de ellos. Los resultados de la tabla 3.5 muestran el porqué la plataforma elegida es Delphi.

CRITERIO DE EVALUACIÓN.	AMBIENTES DE DESARROLLO.		
	Borland C++ Ver. 3.1 para DOS.	Tk/Tcl Ver. Shareware	Delphi Ver. 3.0
Espacio en disco y de memoria requeridos en la instalación del paquete de programación.	Requiere 29.1 Mb de espacio en disco para su instalación compacta.	Requiere de 5.06 Mb de espacio en disco.	Delphi Ver. 3.0 requiere de 30 Mb para su instalación compacta.
Portabilidad.	Puede desarrollar aplicaciones que correrán bajo DOS, Windows 3.0 y Windows 3.1	Puede desarrollar aplicaciones que correrán bajo Windows 3.0, Windows 3.1, Windows 3.11, Windows95, Unix, Mac 8.0.	Puede desarrollar aplicaciones que correrán bajo Windows 3.1, Windows 3.11, Windows NT 3.5, OS/2 Warp, y Windows 95.
Desarrollo de la aplicación.	El tiempo que se empleó en la elaboración de la interfaz prueba fue de 8 hrs.	El tiempo que se empleó en la elaboración de la interfaz prueba fue de 5 hrs.	El tiempo que se empleó en la elaboración de la interfaz prueba fue de 3 hrs.
Interacción de la interfaz con el usuario.	La interfaz que se realizó en gráficos para DOS no resultó muy atractiva al usuario.	La interfaz que se realizó en gráficos para Windows resultó vistosa y agradable al usuario.	La interfaz se realizó en gráficos para Windows resultó vistosa y agradable al usuario.
Dominio del lenguaje de programación.	Se cuenta con experiencia en la programación bajo este lenguaje de programación y se considera un dominio del 65%.	No se tiene experiencia en programar bajo Tk/Tcl y se consideran conocimientos en un 5% de este lenguaje.	No se tiene experiencia en programar en Delphi (Pascal) pero debido a la similitud del lenguaje Pascal con el lenguaje C, y de la semejanza de Delphi con Visual Basic (lenguaje con el que ya se ha tenido experiencia), y se consideran conocimientos en un 35% .
Disponibilidad del lenguaje de programación.	Se cuenta con los discos de instalación y bibliografía sobre C++.	Se dispone de los archivos de ayuda y de demostraciones que acompañan a la instalación del paquete.	Se dispone de la ayuda que acompaña al paquete y bibliografía.
Lenguaje de programación depurable.	Cuenta con la opción de depuración. Además crea el archivo ejecutable automáticamente al compilar.	La depuración resultó difícil y no se obtuvo el archivo ejecutable automáticamente.	Cuenta con la opción de depuración. Además, crea el archivo ejecutable automáticamente con la compilación.

Tabla 3.4 Características de cada ambiente de desarrollo en el que se realizó la interfaz prueba.

En los resultados que muestra la tabla 3.5 se emplearon los siguientes criterios de evaluación:

- E - Excelente.
- A - Aceptable.
- P - Pésimo.

CRITERIO DE EVALUACIÓN.	AMBIENTES DE DESARROLLO.		
	Borland C++ Ver. 3.1 para DOS.	Tk/Tcl Ver. Shareware	Delphi Ver. 3.0
Espacio en disco y de memoria requeridos en la instalación del paquete de programación.	A	E	A
Portabilidad.	A	E	A
Desarrollo de la aplicación.	P	A	E
Interacción de la interfaz con el usuario.	P	E	E
Dominio del lenguaje de programación.	E	P	A
Disponibilidad del lenguaje de programación.	E	A	A
Lenguaje de programación depurable.	E	P	E
PUNTUACIÓN FINAL	OPCIÓN REGULAR.	OPCIÓN REGULAR.	MEJOR OPCIÓN.

Tabla 3.5 Puntuaciones obtenidas por los ambientes de desarrollo considerados para el desarrollo del sistema.

El sistema fue desarrollado en una computadora tipo PC con procesador 486 y velocidad de 50 Mhz., con 8 megabytes en memoria RAM, disco duro de 1.2 gigabytes y monitor SVGA.

3.2 Evaluación del sistema.

En esta sección se discuten algunos resultados del desempeño del sistema. Nuestra experimentación se enfocó a la realización de dos tipos de evaluaciones:

- Una serie de experimentos dedicada a evaluar el desempeño grupal de nuestra colonia.
- Una serie de experimentos dedicada a analizar el desempeño de un individuo de nuestra colonia.

Las condiciones iniciales se refieren a la posición de los agentes-hormiga en la base de datos y al tamaño de la población. La posición de cada uno de los individuos involucrados en los experimentos, fue generada aleatoriamente¹¹ a través de nuestro sistema.

Los resultados que se presentan han sido obtenidos de corridas realizadas en una computadora tipo PC con procesador 486 y velocidad de 50 Mhz., con 8 megabytes en memoria RAM, disco duro de 1.2 gigabytes y monitor SVGA.

3.2.1 Desempeño grupal.

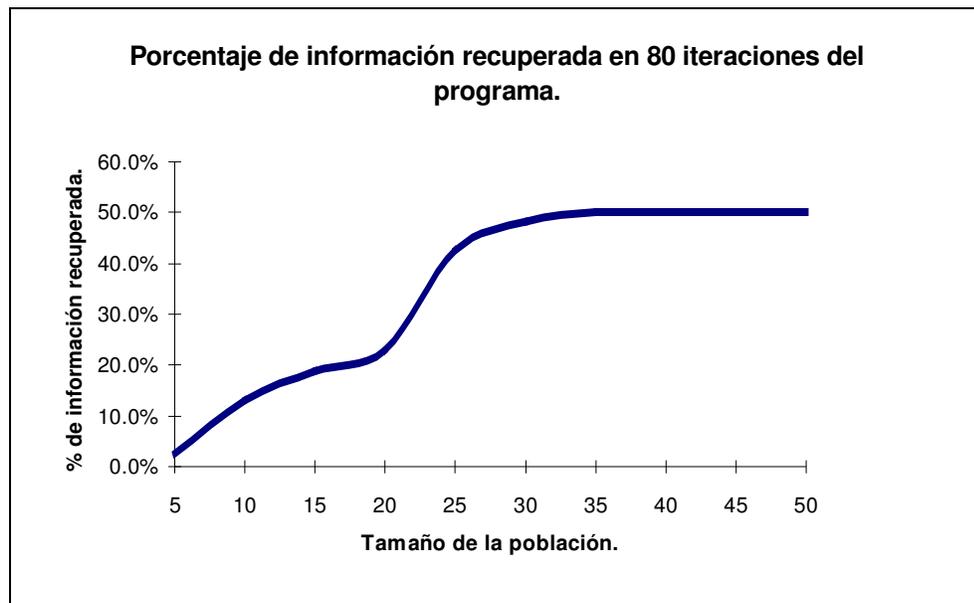
Los primeros experimentos realizados se enfocaron a medir el desempeño grupal de los agentes, para ello se realizó una prueba en donde diferentes grupos de agentes se enfrentaron a solucionar una misma tarea. La tarea consistió en encontrar el término “davis” en una BD distribuida localmente.

Para comparar el desempeño en términos equitativos, a cada uno de los grupos en que se experimentó, se les dejó actuar el mismo tiempo (número de iteraciones). Se fijó un número máximo de iteraciones del programa y se registraron los porcentajes de información recuperada.

¹¹ Para generar la posición aleatoria de los agentes se programó una función que utilizó el *random* del lenguaje (Delphi 3.0).

Las gráficas siguientes ilustran los resultados que se obtuvieron, estas gráficas muestran en el eje x el tamaño de la población, el eje y muestra el porcentaje de la información que fue recuperada, y finalmente el eje z (cuando se grafica) muestra el tiempo en número de iteraciones en que se les dejó actuar.

Se experimentó en grupos de 5, 10, 15, 20, 25, 30, 35, 40, 45 y 50 individuos. En el primer experimentó se les dejó actuar durante 80 iteraciones. La gráfica 3.1 nos muestra cómo una colección de 30 individuos es capaz de recolectar el 50% de información en 80 iteraciones del programa. A partir de este tamaño de la población, si se agregan nuevos individuos el desempeño grupal no se ve favorecido es posible que sólo se pueda recuperar este porcentaje de información en el tiempo indicado como máximo sin que interfiera en ello el tamaño de la población. Por otro lado, en los grupos pequeños el desempeño es menor y se ve incrementando a la par que se incrementa el tamaño de la población hasta llegar a 30 individuos.

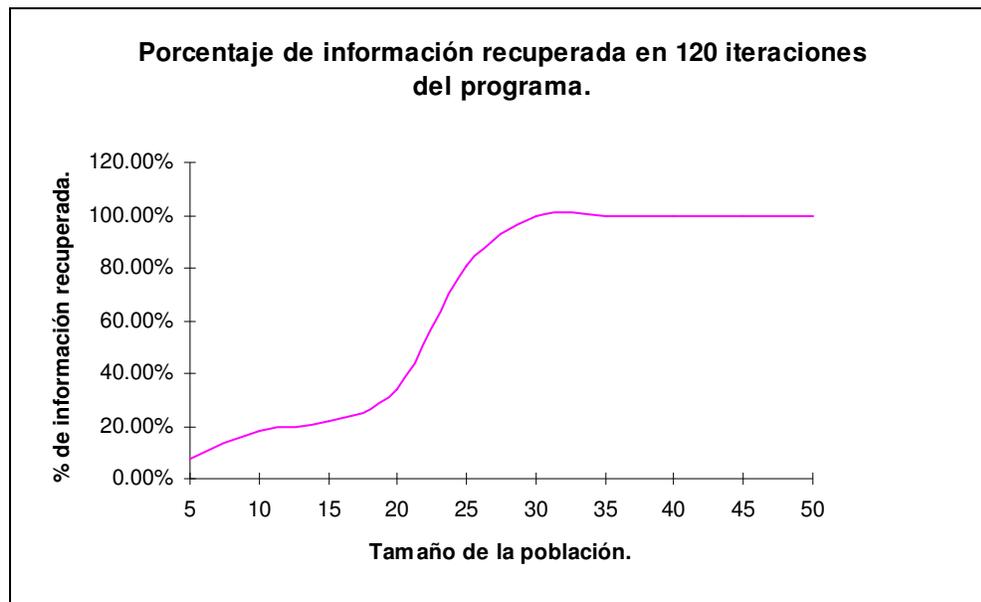


Gráfica 3.1. Resultados obtenidos en 80 iteraciones del programa variando el tamaño de la población.

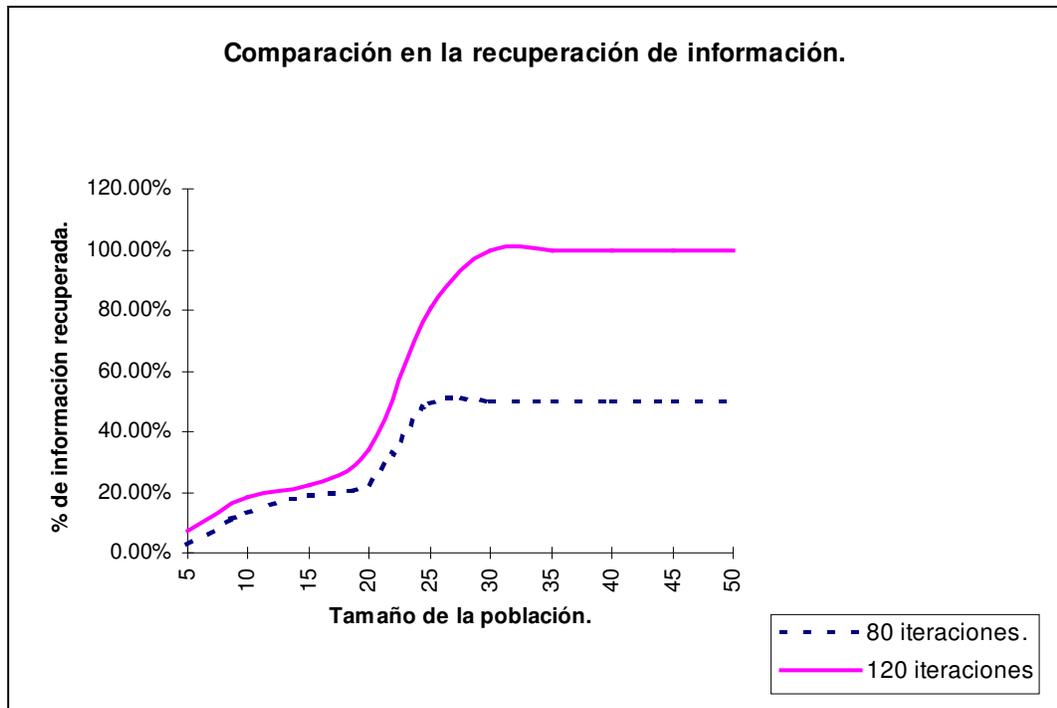
El siguiente experimentó consistió en aumentar a 120 el número de iteraciones para los mismos grupos de agentes-hormiga anteriores, los resultados aparecen en la gráfica 3.2.

La gráfica 3.2. muestra que el 100% de información fue recuperada con una población de 30 individuos en 120 iteraciones. Comparando estos resultados con los resultados de la gráfica 3.1 se observa que sólo se recuperó el 50% de la información debido a que el desempeño grupal estuvo limitado a las 80 iteraciones en que se les permitió actuar a cada grupo.

Para observar mejor las diferencias en el desempeño grupal con diferente número de iteraciones, se realizó una comparación de los dos casos anteriores. La gráfica 3.3. muestra la comparación de la gráfica 3.1 y 3.2.



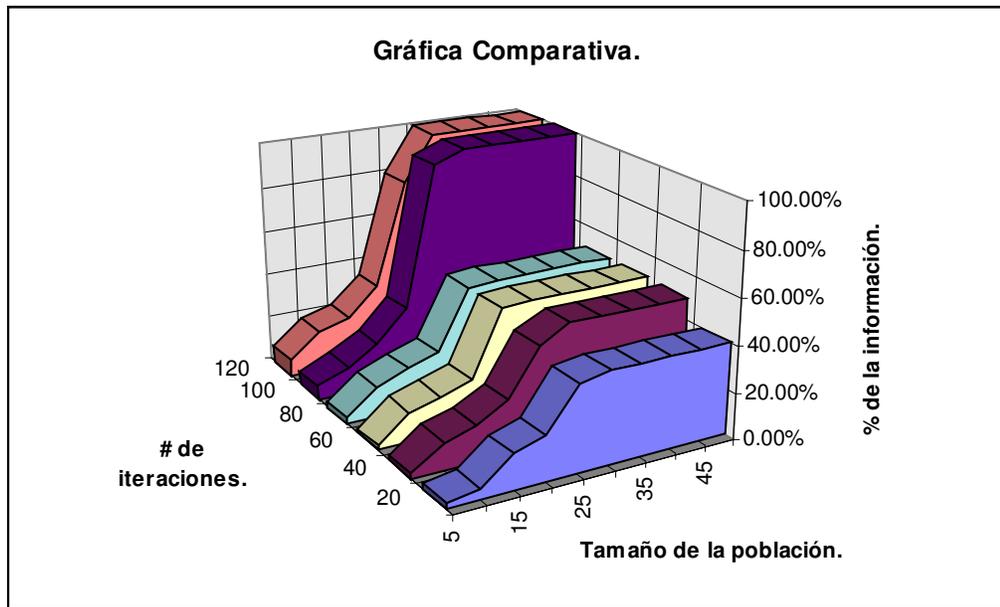
Gráfica 3.2. Resultados obtenidos en 120 iteraciones del programa variando el tamaño de la población.



Gráfica 3.3. Comparación de las gráficas 3.1. y 3.2

La gráfica 3.3 indica que el desempeño grupal está influido por el número de iteraciones en que se les permite actuar. Los grupos de individuos que actuaron por más tiempo presentaron mejor desempeño recuperando mayor cantidad de información con respecto a los grupos de individuos que actuaron menor tiempo. Los individuos que disponen de mayor tiempo para actuar pueden recuperar más información relevante y también pueden marcar la información asociada, de este modo se refuerza el mecanismo de comunicación indirecta consiguiendo un mejor desempeño grupal.

El siguiente experimento consistió en analizar el comportamiento grupal de nuestros agentes-hormiga para diferente número de iteraciones. Los resultados se muestran en la gráfica 3.4.



Gráfica 3.4. Comparación del porcentaje de información recuperada variando el número de iteraciones del programa y el tamaño de la población.

De la gráfica 3.4 observamos que para grupos pequeños de 5, 10, 15, 20 y 25 individuos el desempeño grupal no varía en demasía para los diferentes números de iteraciones. Muestra también que el porcentaje de información recuperado es menor a menor cantidad de iteraciones para los distintos grupos de individuos. Posteriormente el porcentaje de información recuperada no sobrepasa el 50%, para 40, 60 y 80 iteraciones del programa. Y finalmente en 100 y 120 iteraciones se consigue recuperar el 100% de la información. Para 100 iteraciones del programa hay un incremento significativo en cuanto a la recolección de información obtenida por los individuos, este suceso es un indicativo de que el desempeño grupal se ve afectado por la distribución de la información en de la BD. Por otro lado en 120 iteraciones hay una mejoría en el desempeño de los grupos pequeños, sin embargo a partir del grupo de 25 individuos el incremento del porcentaje de la información recuperada se hace menor que el porcentaje recuperado en 100 iteraciones por los mismos grupos, es posible que esto se deba a que con 100 iteraciones se alcanza a recuperar el 100% de la información y con 120 iteraciones se produce un exceso de información que impide que los individuos trabajen ordenadamente.

Finalmente se estudio si el sistema es sensible a la manera en cómo se distribuyen los agentes-hormiga en el área de búsqueda. Se instrumentaron dos formas de exploración, cómo se mencionó antes el mecanismo de distribución de los agentes-hormiga es aleatorio. Estas diferentes formas de exploración varían en el grado de aleatoriedad que se emplea en la distribución de los individuos. La descripción de estas formas de exploración es la que sigue:

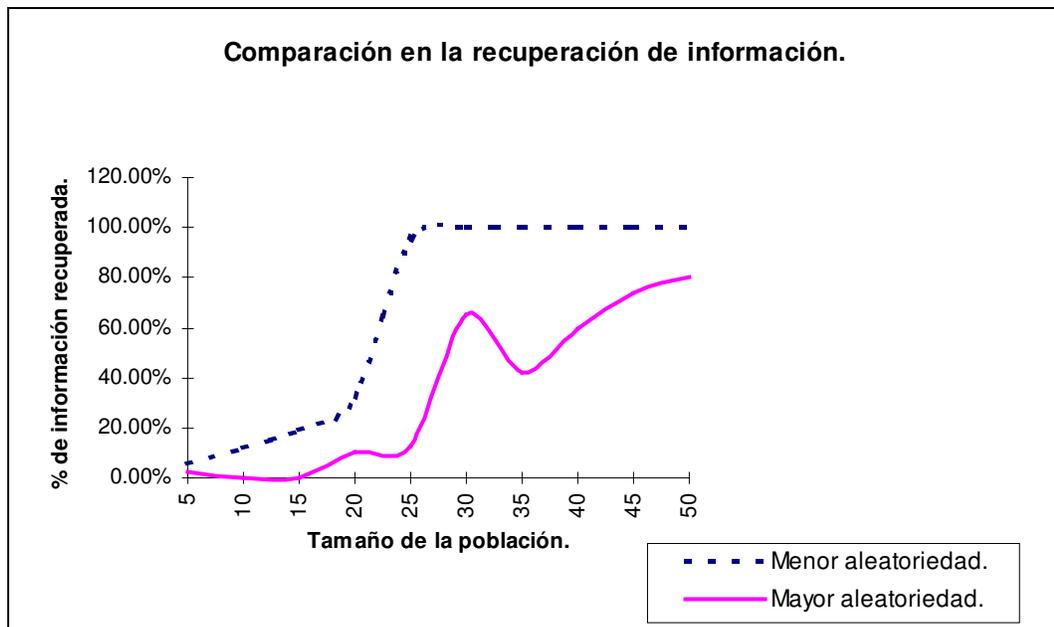
Consideremos que la BD distribuida (BDD) sobre la que nuestros individuos van a explorar se encuentra fragmentada en dos, designaremos como BD1 al primer fragmento y cómo BD2 al segundo fragmento.

1. Distribución con menor aleatoriedad. Esta forma de distribución consiste en que los primeros agentes-hormiga se distribuyen aleatoriamente en BD1 y después se distribuyen aleatoriamente los siguientes agentes-hormiga en BD2. Si bien la distribución es aleatoria, se considera primero la exploración de un fragmento y posteriormente se explora el otro.
2. Distribución con mayor aleatoriedad. Esta distribución consiste en asignar la posición de cada agente-hormiga indistintamente en BD1 o en BD2.

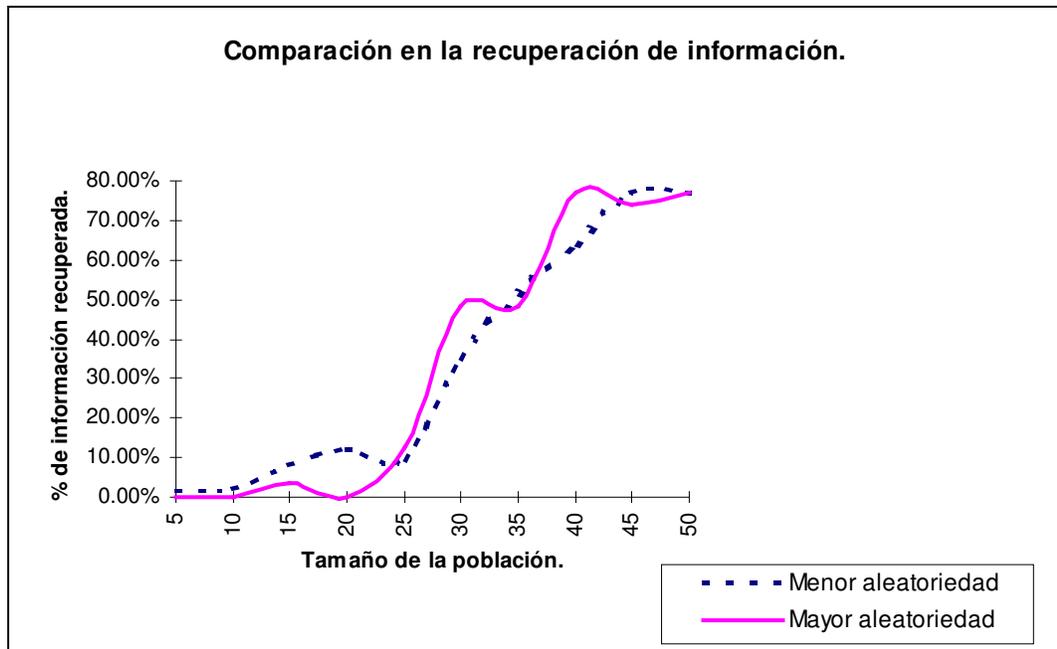
La gráfica 3.5. muestra los resultados de emplear ambos mecanismos de distribución de los agentes-hormiga. La tarea a alcanzar es la misma que la de los experimentos anteriores (encontrar el término “davis”), la prueba se realizó para los mismos grupos y el número de iteraciones que se les permitió actuar fue de 100.

Los resultados indican que se requiere de un número mayor de agentes-hormiga para cumplir con la tarea si se emplea mayor aleatoriedad en la distribución. Con el mecanismo de menor aleatoriedad de consigue recuperar el 100% de información con una población de 30 agentes-hormiga, mientras que si se emplea una distribución de mayor aleatoriedad no se alcanza a recuperar este porcentaje, y sólo se recupera en 100 iteraciones el 80% de información. El sistema mostró ser sensible a la distribución de los agentes. Sin embargo el término “davis” se localizaba los primeros campos de BDD. ¿Qué ocurriría si el dato se localizaba en los últimos campos de BDD?, para responder a esta pregunta se aplicaron las mismas condiciones que en este

experimento a la solución de otra tarea, que consistía en encontrar el término “buy” (el cual se localizaba en los últimos campos de BDD), los resultados se observan en la gráfica 3.6 e indican que sólo se alcanzó a recuperar el 80% de información indistintamente del mecanismo de distribución. El número de hormigas necesarias para ello no varía significativamente para ambos mecanismos de distribución. Comparando ambos casos, tenemos que el desempeño de grupal varía en mayor grado si hay mayor aleatoriedad en la distribución y el dato a encontrar se encuentra en los primeros campos de BDD, y el desempeño grupal no varía significativamente si la información a recuperar se encuentra en los últimos campos de BDD.



Gráfica 3.5. Comparación en la recuperación de información relacionada al término ‘davis’ empleando diferentes mecanismos de distribución de los agentes.



Gráfica 3.6. Comparación en la recuperación de información relacionada al término ‘buy’ empleando diferentes mecanismos de distribución de los agentes.

En las gráficas 3.5 y 3.6 se observa que hay mayor variabilidad en el comportamiento de la sociedad. Las gráficas indican que a mayor aleatoriedad se tienen mayor variabilidad y menos predictibilidad.

3.2.2 Desempeño de un individuo.

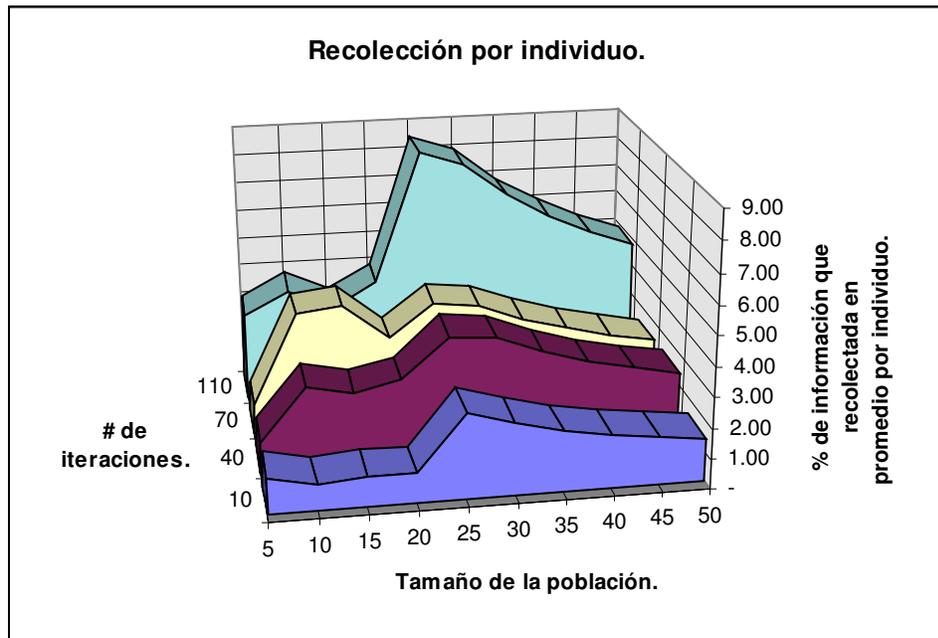
En esta sección se intenta demostrar el resultado holístico del sistema, esto es que “El resultado del todo es más que la suma de las partes”. El objetivo es demostrar que un agente-hormiga que trabaja en un grupo consigue más que si trabajase solo, pues se ve beneficiado del esfuerzo de los demás, su eficiencia emerge de la colaboración con los demás agentes-hormiga.

Para medir el desempeño de un agente-hormiga se estudia el rendimiento que tiene un individuo al enfrentarse a tareas semejantes, pero colaborando en grupos de distinto tamaño. La tarea que solucionaron nuestros agentes-hormiga fue encontrar la palabra “davis”, se experimento en grupos de 5, 10, 15, 20, 25, 30, 35, 40, 45 y 50 individuos, utilizando el

mecanismo de comunicación indirecta, se les permitió actuar durante 10,40,70 y 110 iteraciones respectivamente.

El desempeño promedio de cualquier agente-hormiga del grupo, se obtuvo calculando el porcentaje promedio de información recuperada en promedio por individuo. Este promedio se obtiene como la razón entre el número total de información encontrada por el grupo y el número de individuos que forman el grupo. El parámetro fijo fue el número de iteraciones en que se les permitió actuar.

La gráfica 3.7 el desempeño por individuo para los diferentes grupos.



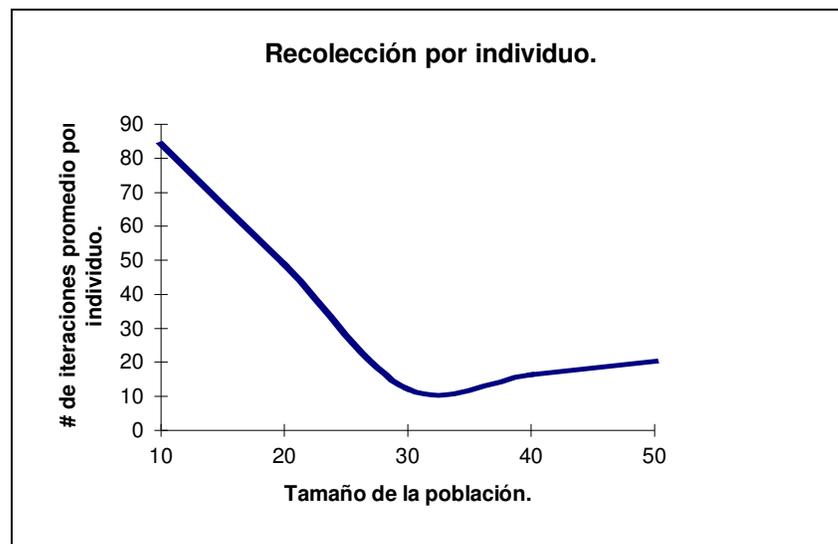
Gráfica 3.7. Promedio de información recolectada por individuo.

La gráfica 3.7 muestra que para diferente número de iteraciones, en un grupo de 25 individuos se alcanza mejor desempeño por individuo. Esto es, en una población de tamaño 25 se recolecta mayor cantidad de información en promedio por individuo.

En 110 iteraciones en un grupo de 25 individuos se obtuvo el mejor desempeño. Para mayor cantidad de individuos el desempeño se ve

decrementado, la población no se beneficia al aumentar la cantidad de agentes que en lugar de “ayudarse” se “estorban”. Los resultados de la gráfica 3.7 muestran que si sostiene que el “todo es más que la suma de las partes”. De no ser así como resultado tendríamos una función lineal a graficar, se esperaría que a mayor cantidad de agentes-hormigas el promedio de información recolectada por individuo fuese menor, sin embargo la gráfica nos muestra que un agente-hormiga dentro del grupo se ve beneficiado del esfuerzo de las demás. Como resultado del mecanismo de comunicación indirecta empleado en nuestros agentes hay *algo* que emerge y se observa en su desempeño.

Se analizó el desempeño individual también a partir del tiempo promedio por individuo, se graficaron el número de iteraciones promedio que le toma a un individuo de un grupo cumplir con una tarea. La tarea que se fijó fue la recuperación del 100% de información para el término “davis”. Se experimentó en grupos de 10, 20, 30, 40 y 50 individuos. Este promedio se obtuvo como la razón, entre el número de interacciones en total que le toma al grupo finalizar una tarea entre el número promedio de unidades recolectada por cada individuo (Muñoz 1997). Los resultados se muestran en la gráfica 3.8.



Gráfica 3.8. Promedio de iteraciones por individuo.

Estos resultados indican que para la población de 30 individuos se obtiene un mejor desempeño, esto es, se obtiene un menor número de

iteraciones en promedio por individuo para recolectar una unidad de información.

3.3.Comparación del sistema con algunos métodos de búsqueda tradicionales.

Los experimentos realizados en la sección 3.2 son adecuados para demostrar el desempeño de nuestra colonia artificial. Sin embargo es necesario establecer una comparación con los métodos tradicionales de búsqueda y recuperación de información.

La comparación se enfocó a evaluar los siguientes parámetros:

- a) Uso de recursos (tiempo máquina).
- b) Forma en que se presentan los resultados.
- c) Porcentaje de exactitud en los resultados.
- d) Opciones para personalizar la búsqueda (permite utilizar operadores lógicos como OR, AND. O bien, si permite elegir los campos que se mostrarán en los resultados).
- e) Si permite búsquedas más versátiles (Si encuentra sinónimos a lo que el usuario busca).

Los resultados que se muestran se obtuvieron al realizar las pruebas en una computadora tipo PC con procesador pentium y velocidad de 166 Mhz., 15 Mb en memoria RAM, disco duro de 1.5 Gb. y monitor Super VGA.

La BD sobre la que se realizaron las pruebas de búsqueda está en formato dBase y su dimensión es de 10320 registros y 24 campos.

Los métodos de recuperación de información que se utilizaron son:

- A)Access'97.
- B)dBase III plus Ver. 1.0 IBM/MSDOS.
- C)Secuen.exe (Programa realizado en Borland C++ Ver. 3.1, para realizar búsquedas secuenciales).

D)Hormigas.exe (Sistema multi-agente realizado en Delphi 3.0. Este sistema es la implementación del diseño descrito al inicio del capítulo).

Para realizar una evaluación equitativa, a cada método de recuperación anterior se le solicitó cumplir con la tarea de localizar la palabra ‘baja’ y se le indicó el campo donde se deseaba localizar la información, el nombre de este campo es CLASS. Los resultados se muestran a continuación.

3.3.1 Resultados de la Comparación.

A) Access’97.

- El tiempo promedio invertido en realizar la consulta es igual a 25 segs. (Considerando el tiempo invertido en el diseño de la consulta).
- Los resultados son mostrados en un Grid. Para mostrar información cuenta son un asistente. Con la ayuda de este asistente el usuario puede visualizar la información de los campos que desea. Sin embargo el asistente no auxilia al usuario a visualizar los resultados de la búsqueda de un dato en particular. Para realizar búsquedas es necesario manejar instrucciones en SQL (Es necesario que el usuario conozca SQL).
- La exactitud de resultados es del 100% si el dato se encuentra en la base de datos escrito en mayúsculas o minúsculas, pero los resultados no son exactos si el dato se encuentra en la base de datos fragmentado por espacios u oculto (es necesario utilizar comodines en la instrucción SQL)
- Permite personalizar la búsqueda al utilizar operadores lógicos como OR, AND además de permitir seleccionar los campos a mostrar en los resultados (Con la ayuda del asistente o mediante una instrucción SQL).
- No permite la recuperación de información sinónima, a menos que se les especifique explícitamente cada sinónimo que se desea encontrar mediante una instrucción SQL.

B) dBase III plus.

- El tiempo promedio invertido en la búsqueda es de 1 minuto (mostrando los resultados en forma de lista, son desplegados rápidamente sin pausa). El tiempo promedio invertido en la búsqueda, si los datos se muestran pausadamente es de 15 minutos.
- Los resultados se presentan separados por espacios, es probable que cree posible confusión al usuario.
- Las búsquedas son diseñadas por el usuario con la ayuda de menús. Encuentra Datos Precisos en un 100%. No encuentra Datos Equivalentes ó Datos Similares. Para el caso de los Datos ocultos, encuentra datos en una frase sólo si la palabra a buscar se localiza al inicio de la frase y está tal y cómo la busca el usuario. Si se trata de encontrar el dato dentro de la frase no es localizado.
- Permite combinar el uso de AND Y OR, y permite elegir los campos a mostrar y los campos sobre los cuales se realizará la búsqueda.
- No permite la recuperación de información sinónima.

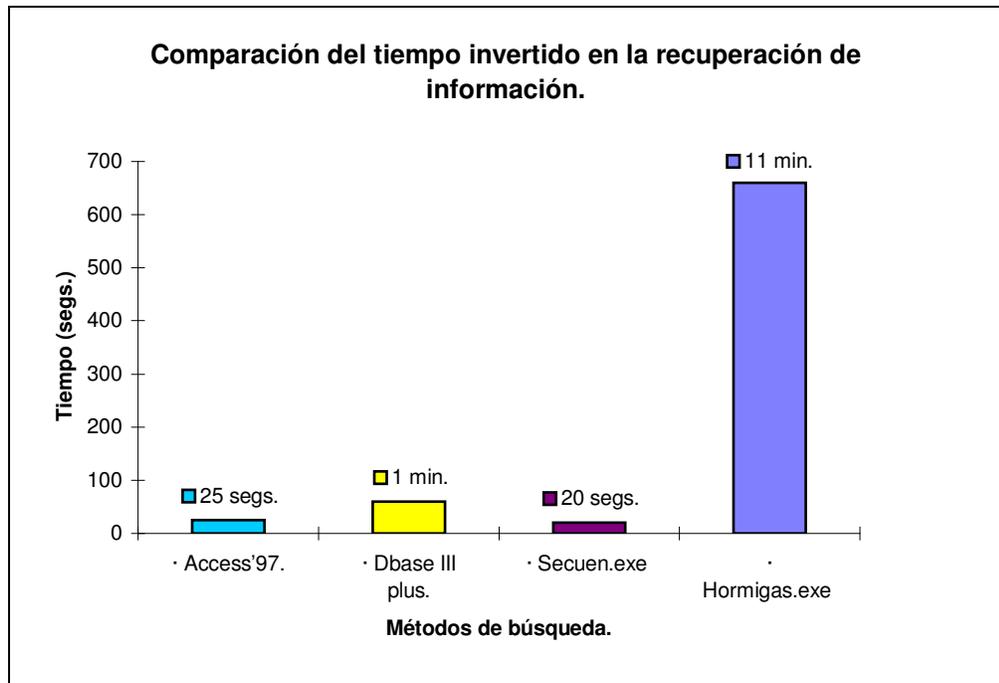
C) Secuen.exe

- Tiempo promedio invertido en la búsqueda es de 20 segundos.
- Despliega los resultados en forma de lista.
- El usuario debe conocer el número de campo dónde posiblemente se encuentre el dato a buscar. Sólo encuentra los Datos Precisos y los Datos Equivalentes, no encuentra Datos Similares ni Datos ocultos.
- Una restricción del programa es que no permite el uso de AND y OR. No permite elegir los campos a mostrar en los resultados, sólo busca en un campo específico y muestra sólo los resultados de dicho campo.
- No permite búsquedas con sinónimos.

D) Hormigas.exe

- El tiempo promedio invertido en la búsqueda es de 11 minutos empleando una sólo conducta, con una único agente-hormiga y buscando información en un único campo.
- Muestra los resultados en un Grid.
- Encuentra Datos Precisos, Datos Equivalentes, Datos Similares ó Datos ocultos en un 100%. Puesto que permite localizar sinónimos también encuentra Datos Sinónimos Precisos, Datos Sinónimos Equivalentes, Datos Sinónimos Similares ó Datos Sinónimos ocultos en un 100%.
- Al igual que el programa anterior tiene la restricción de no permitir el uso de AND y OR. Permite elegir los campos sobre los cuales realizar la búsqueda, mismos que se mostrarán en los resultados. También ayuda en la personalización de la búsqueda al permitir al usuario elegir la conducta o conductas que el agente empleará durante la ejecución del sistema. Es decir, permite elegir el tipo de dato que se desea encontrar, si se desea buscar un dato preciso, equivalente, similar, oculto, sinónimo preciso, sinónimo equivalente, sinónimo similar, ó sinónimo oculto.
- Permite búsquedas con sinónimos.

La tabla 3.6 nos muestra un resumen de lo anteriormente explicado y la gráfica 3.9. nos indica el tiempo que le llevó a cada método mostrar los resultados al experimento descrito al inicio de la sección.



Gráfica 3.9. Comparación del tiempo invertido en la búsqueda del término 'baja' para diferentes métodos de búsqueda.

MÉTODOS DE BÚSQUEDA	PÁRAMETROS A EVALUAR				
	Uso de recursos	Presentación de resultados	Exactitud en los resultados	Si Permite personalizar la búsqueda.	Si encuentra sinónimos del dato a buscar.
Access'97	25 segs.	Se muestran en un <i>Grid</i> .	100% para el caso de Dato Preciso y Dato Equivalente. Se presenta inexactitud en los resultados mostrados para los casos de Dato Similar y Dato Oculto.	Permite el uso de AND y OR en la construcción de las búsquedas. Permite elegir los campos que se desean mostrar en el resultado de la búsqueda.	No.
dBase III	60 segs ó 1 minuto.	Se muestran separados por espacios.	100% en el caso de Dato Preciso. Para Dato Equivalente, Dato Similar no los muestra. El Dato Oculto sólo se muestra bajo determinadas restricciones.	Permite el uso de AND y OR en la construcción de las búsquedas. Permite elegir los campos que se desean mostrar en el resultado de la búsqueda.	No
Secuen.exe	20 segs.	Se muestran en forma de lista.	100% en el caso de Dato Preciso y Dato Equivalente. El Dato Similar y el Dato Oculto no se muestran.	No permite el uso de AND y OR en la construcción de las búsquedas. Ni permite elegir los campos que se desean mostrar en el resultado de la búsqueda.	No
Hormigas.exe	11 minutos	Se muestran en un <i>Grid</i> .	100 % para cualquiera de los casos: Dato Preciso, Dato equivalente, Dato Similar y Dato Oculto.	No permite el uso de AND y OR en la construcción de las búsquedas. Pero si permite elegir los campos que se desean mostrar en el resultado de la búsqueda.	Sí

Tabla 3.6. Resumen de los resultados de la comparación del sistema multi-agente con algunos métodos de búsqueda tradicionales.

Los resultados cómo puede observarse son un indicativo de que nuestro sistema es más lento que los métodos tradicionales de búsqueda aquí presentados.

A pesar de que estos resultados mostraron un rendimiento diferente al esperado para solucionar la tarea de recuperación de información en una BD, no implica que debe descartarse el uso de las Metodologías Basadas en el Comportamiento para la solución de problemas complejos.

Si bien se mostró que los agente-hormiga son capaces de organizarse y conseguir la recuperación global de información, su implementación también mostró problemas en la ejecución del sistema en tiempo real. En la práctica, el problema en el cual estamos interesados es suficientemente difícil que los agentes, aún si ellos cooperan, es improbable que encuentren la solución en sólo unos cuantos pasos.

Para Jennings & Wooldridge (Jennings & Wooldridge 1998) la tecnología de agentes se considera útil si puede ofrecer la habilidad para solucionar problemas en una mejor forma (más barata, más fácil, más eficiente, o más rápido). La arquitectura del sistema multi-agente que fue propuesto en esta disertación, ofreció solucionar el problema de búsqueda y recuperación de información de una forma más barata en el sentido de que los agentes que se propusieron son agentes reactivos, con bajo nivel cognoscitivo, con un diseño relativamente simple, sin programación compleja y que mediante el uso de mecanismos de comunicación elementales, son capaces de colaborar para resolver el problema que nos ocupa; a partir de una programación simple se puede resolver lo no simple. Sin embargo se reconoce que los sistemas reactivos están entre los tipos de agentes más complejos de diseñar e implementar. Para ciertos tipos de sistemas reactivos, aún técnicas de ingeniería de software y herramientas especializadas fallan, se requieren de nuevas técnicas (Jennings & Wooldridge 1998).

La mayoría de las aplicaciones que actualmente usan agentes podrían ser construidas sin usar técnicas de agentes. No siempre una solución basada en agentes es la más apropiada, o incluso es factible. Se tienen problemas comunes a todas las aplicaciones basadas en agentes (Jennings & Wooldridge 1998), uno de ellos es precisamente la respuesta en tiempo real. Una solución basada en agentes podría no ser apropiada para dominios donde la respuesta

en tiempo real debe ser garantizada. Un ejemplo de ello son los agentes personales, que no son más rápidos que las máquinas de búsqueda comunes, así que el usuario sacrifica velocidad por calidad de la información (Lesser et. al. 1998).

Además la metáfora del hormiguero en la se inspiró nuestro sistema ofrece una solución distribuida al problema (Langton et. al. 1995). Lander en (Lander 1994) indica que en una búsqueda distribuida se tienen múltiples elementos que pueden simultáneamente desarrollar búsquedas locales en diferentes partes del dominio del problema; los resultados parciales de las búsquedas locales son compartidos en alguna forma así que la respuesta deseada se produce en una manera oportuna. Sin embargo una efectiva búsqueda distribuida podría no ser óptima para una búsqueda secuencial.

Como antecedentes de la evaluación a la búsqueda cooperativa se tiene:

El trabajo de Hogg & Williams (Hogg & Williams 1993) que se presenta y evalúa experimentalmente la hipótesis de que la búsqueda cooperativa paralela es adecuada para los problemas difíciles. Este documento nos dice que un grupo de agentes cooperativos pueden ofrecer solucionar tales problemas más rápido que el mismo número de agentes trabajando independientemente. Los agentes descritos en (Hogg & Williams 1993) utilizan marcas para comunicarse y de esta forma cooperan en la solución del problema. Leen y escriben las marcas en un blackboard común. Los resultados experimentales presentan datos basados en ejecuciones no optimizadas en tiempo, los experimentos de las búsquedas se realizaron en una implementación serial en los cuales los pasos individuales son multiplexados (Cada agente en el grupo realiza un único paso, este procedimiento se repite hasta que un agente encuentra una solución). Los resultados mostraron que la búsqueda cooperativa con respecto a los agentes independientes es más lenta, se propone que una implementación paralela del grupo de agentes disminuiría el tiempo de ejecución, los agentes deberían relativamente realizar más pasos en una ejecución en paralelo que en una ejecución serial. Se sugiere que los métodos cooperativos pueden ser probablemente beneficiados por la solución paralela de problemas muy difíciles.

En (Clearwater et. al. 1992) se presenta una evaluación cuantitativa del valor de la cooperación para solucionar problemas de verificación de restricciones a través de una serie de experimentos, así como una teoría general de la solución cooperativa del problema. Estos experimentos exhiben claramente un mejoramiento universal en el desempeño que resulta de la cooperación. La cooperación conduce a mejoramientos en el desempeño de un grupo de individuos. Los resultados sugieren una metodología alternativa para las técnicas existentes para solucionar satisfactoriamente problemas en la computación e inteligencia artificial distribuida. Se propone que un grupo de agentes que cooperan puede solucionar un problema o una tarea más rápido que un sólo agente o él mismo grupo de agentes trabajando aisladamente. Los resultados mostraron que el desempeño de los agentes que colaboran en los problemas de verificación de restricciones es altamente incrementado cuando se compara con el desempeño del mismo grupo de agentes que no interactúan. En los experimentos de los agentes cooperativos, los agentes se comunican a través de marcas que son escritas en un blackboard central. La búsqueda cooperativa permite que el agente use marcas encontradas por otros agentes. En la práctica, el problema de verificación de restricciones es lo suficientemente difícil que los agentes, aún cooperando, no encuentran una solución en sólo unos cuantos pasos, se propone dirigir la distribución de la cooperación con una búsqueda paralela. Sea N el número de agentes o procesos involucrados en la búsqueda, en un sistema completamente paralelo, cada uno de estos puede corresponder a un procesador separado, pero ellos podrían ser simulados por un pequeño número de procesadores también. (Clearwater et. al. 1992).

En (Oates et. al. 1994) se explica el paradigma de la Recuperación Cooperativa de Información. Este paradigma involucra acceso concurrente y asíncrono a una red de información por un grupo de agentes inteligentes. El modelo de Recuperación Cooperativa de Información ve a la recuperación de información como una actividad que soluciona un problema distribuido, este problema se caracteriza por la existencia de interdependencias entre los subproblemas, conduciendo a la necesidad de coordinar extensivamente a los agentes durante la solución del problema (Decker et. al. 1995). Se intenta dar una solución distribuida al problema de la adquisición cooperativa de la información. De modo que las redes de información distribuida a gran escala no sólo necesitan incrementar el paralelismo en la búsqueda sino que también motivan a la recuperación cooperativa y a la construcción dinámica de

respuestas a queries. El documento proporciona un modelo de Recuperación Cooperativa de la Información como un proceso de solución distribuida al problema y que se auxilia de los métodos existentes en sistemas multi-agentes. Se sugiere que existen métodos en los sistemas multi-agentes que pueden servir para ofrecer ventajas en futuras implementaciones de los sistemas recuperadores de información basados en esta perspectiva.

En (Clearwater et. al. 1992) se presenta la teoría de la cooperación entre agentes, se da una explicación simple del fenómeno en general y lo más importante es que se sugiere emplear la cooperación entre agentes no sólo para la solución del problema de verificación de restricciones, que es el que se aborda en particular, sino que plantea emplear la cooperación entre agentes de un modo más general en la solución de otra clase de problemas.

Las investigaciones anteriores difieren con nuestro trabajo en el problema que desean solucionar con la ayuda de agentes cooperativos así como en la arquitectura de los agentes. Se asemeja a nuestra propuesta en que el grupo de agentes tratan de solucionar su problema particular colaborando entre ellos, comunicándose vía un blackboard central. Estudian el fenómeno de la cooperación entre agentes de un modo más genérico, derivan de los resultados que obtienen que la cooperación entre agentes puede emplearse como una herramienta en la solución a problemas difíciles. Los mecanismos relativamente simples que son capaces de dar un desempeño incrementado sugiere que tales métodos podrían ser muy fácilmente aplicados (Hogg & Williams 1993).

Otro punto en el que se asemeja a nuestra propuesta consiste en que la implementación de los agentes cooperativos no resulta óptima en tiempo real. Y proponen emplear una arquitectura paralela para hacer eficiente en tiempo la implementación de estos agentes.

Pues bien, el diseño de la arquitectura de los agentes-hormiga es inherentemente paralelizable (Deneubourg et. al. 1991, Deneubourg et. al. 1992). Los agentes podrían estar trabajando simultáneamente en soluciones alternativas para solucionar el problema cómo lo sugiere Lander en (Lander 1994). La distribución del problema implica su descomposición en subproblemas para ser solucionado por el procesamiento de múltiples unidades tales cómo CPU's o agentes (Oates et. al. 1994).

En el reino de la computación la emergencia de máquinas masivamente paralelas subrayan el poder asumido de la concurrencia para la solución de problemas muy complejos que pueden ser descompuestos en pequeñas piezas, y se ha iniciado un gran esfuerzo dedicado al diseño de algoritmos paralelos para la solución de problemas computacionalmente difíciles. Cuando los problemas son lo suficientemente difíciles que muchos pasos se requieren para solucionarlo, aún con la cooperación. Por simplicidad, se supone que cada agente corre en su propio procesador. La forma más natural de pensar sobre una colección de procesos independientes es que se ejecuten posiblemente en procesadores separados. Sin embargo, es posible multiplexar entre los procedimientos seguidos por un conjunto de agentes (Clearwater et. al. 1992).

La historia reciente ha mostrado que las computadoras paralelas pueden ser aprovechadas para solucionar eficientemente problemas complejos, pero óptimamente implementar tales programas puede ser difícil para los programadores humanos (Ray & Thearling 1997).

Para nuestro proyecto aún quedaría por responder a la pregunta de qué tan diferente serían los resultados en una implementación en paralelo, al igual que para el trabajo realizado en (Hogg & Williams 1993).

Finalmente, de una manera más general, cuando se aplica la optimización a problemas, uno debería necesitar considerar la calidad de las soluciones obtenidas tanto como el tiempo requerido (Hogg & Williams 1993).

3.4. Trabajos relacionados.

El número de experimentos sobre el comportamiento grupal de agentes inspirados en insectos sociales ha sido extenso, los objetivos de estos experimentos son diversos, los hay dirigidos a la investigación etológica así como también los hay en el diseño de algoritmos de propósito específico.

Se tienen estudios etológicos de las habilidades mostradas por las sociedades de insectos para el forrajeo colectivo (Drogoul & Ferber 1993;

Muñoz 1997) y la auto-organización espacial y conductual (Ünsal 1993), entre otros.

También se analiza a la auto-organización, al comportamiento emergente global complejo y a la habilidad para construir nidos (estructuras complejas) como habilidades a considerar en los procesos empleados por los insectos sociales, y que se presentan como una alternativa en la construcción o producción de artefactos (Deneubourg et. al. 1992).

La metáfora del hormiguero en particular ha tenido diferentes aplicaciones, una de ellas es la propuesta de una arquitectura no jerárquica, no centralizada capaz de generar un comportamiento global complejo (Langton et al. 1995), o bien como un algoritmo de ordenamiento distribuido (Deneubourg et al. 1991), también se ha empleado como propuesta para solucionar el problema del agente viajero (Colorni et. al. 1992).

Por otro lado trabajos que no se han inspirado en colonias de insectos sociales pero que son dignos de considerarse por mostrar otra forma de solucionar el problema de búsqueda y recuperación de información. Estos trabajos están orientados principalmente a la búsqueda y recuperación de información en el World Wide Web (Web en adelante). El Web está creciendo rápidamente y encontrar información útil en este medio es una tarea en la que los usuarios invierten mucho tiempo; además, la complejidad de este ambiente demanda de la realización de programas que muestren un nuevo estilo de interacción entre el usuario y la máquina. Estas propuestas están orientadas a la creación de sistemas que colaboraren de manera “inteligente”, activa y personalizada con los usuarios, ayudándoles en esta difícil tarea. Por un lado están los sistemas que no utilizan agentes en su diseño como son, hasta ahora, los buscadores del Web comunes como Yahoo, Lycos, Infoseek, Alta Vista, Excite, entre otros. La forma en que estos trabajan es similar al buscador llamado Veronica (Foster 1994) que construye consultas en SQL utilizando tipos predefinidos de búsqueda, el usuario utiliza un número especial de opciones para mejorar sus búsquedas. Cabe mencionar a los Web crawlers (Blum et. al. 1998) también conocidos como “spiders”, robots o deambuladores, estos programas automáticamente recorren el Web. Un “crawler” reside en una única máquina al ejecutarse. Generalmente son usados para crear índices que son usados por los buscadores del Web que mencionamos antes. Un “crawler” construye un índice de la página que le fue

consultada, siempre mantiene actualizada la lista de páginas que ha encontrado y añade nuevas páginas cuando las encuentra.

Por otro lado, se cuenta con sistemas basados en agentes. Hay sistemas que utilizan agentes basados en el conocimiento como en (Martínez 1997), para realizar búsquedas en bibliotecas digitales¹² con la ayuda de dos tipos de agentes, agentes de consulta y agentes de visualización, los primeros recogen la consulta del usuario y la buscan en un módulo llamado “indexador” utilizando metainformación¹³, y los segundos toman los resultados de las consultas y los muestran al usuario.

En adelante fijaremos nuestra atención y abundaremos en los sistemas que utilizan agentes inteligentes, que es el enfoque de nuestro interés, y que contienen con problemas semejantes respecto al tratamiento de la información en el Web.

Se tienen por un lado, sistemas donde un único agente colabora con el usuario en el mismo ambiente de trabajo, este tipo de agentes son llamados agentes interfaz, estos son programas que utilizan técnicas de IA y se basan en la metáfora del asistente personal. Un ejemplo de tales sistemas lo representa el agente interfaz de Maes (Maes 1994) que puede aprender el perfil de su usuario y adaptarse a él para facilitarle el trabajo y liberarlo de la sobrecarga de información. Agentes de este tipo han sido propuestos para filtrar noticias electrónicas (Usenet Netnews) así como para recomendar libros, música u otras formas de entretenimiento.

Otro agente interfaz basado en el comportamiento es “Letizia” (Lieberman 1995), este agente asiste al usuario explorando el Web. El usuario trabaja con un explorador tal como Netscape, el agente sigue el comportamiento del usuario e intenta anticiparse a sus intereses haciendo una exploración concurrente y autónoma de las ligas a partir de la posición actual del usuario. “Letizia” carga las ligas de la página actualmente visualizada y a partir de éstas selecciona las páginas que son más compatibles con el perfil

¹² “Por bibliotecas digitales, se entiende a las colecciones de documentos digitales, que pueden estar más o menos organizadas, y donde los elementos pueden consistir en información uniforme (ficheros de texto, referencias bibliográficas, *news*, páginas HTML, imágenes en diversos formatos) como una mezcla de todos o algunos de ellos” (Martínez 1997).

¹³ La metainformación o metadato son informaciones que, sin ser parte de la documentación original, aportan un conocimiento valioso para facilitar su mensaje (Martínez 1997).

del usuario, el cual es aprendido al observar las decisiones del usuario. “Letizia” puede filtrar y recuperar información trabajando en paralelo con el usuario, los resultados son presentados continuamente como recomendaciones al usuario.

Una extensión de “Letizia” es un agente llamado “Let's Browse” (Lieberman et. al. 1997), al igual que “Letizia” filtra las páginas candidatas a través de perfiles aprendidos al observar la actividad del usuario. La diferencia es que no asiste a un único usuario, sino que asiste a un grupo de personas al explorar en el Web, sugiriéndoles nuevo material que probablemente será del interés común. La idea de “Let's Browse” consiste en que los usuarios comparten un monitor. “Let's Browse” sensa la presencia de los usuarios sin ninguna acción explícita de ellos y muestra las páginas del Web que considera puede ser de interés común entre los participantes. “Let's Browse” percibe a los participantes por medio de placas electrónicas que les son colocadas. Estas placas transmiten la identidad del usuario a través de luz infrarroja. El receptor de la luz infrarroja está colocado sobre el monitor donde se ejecuta “Let's Browse”. Las identidades de los usuarios son útiles para indizar los perfiles de los participantes y así obtener un perfil de grupo, estos perfiles deben almacenarse previamente en una base de datos. El sistema aprende el perfil de cada usuario utilizando las mismas técnicas aplicadas en “Letizia”.

“WebMate” (Chen & Sycara 1997) ayuda a explorar y a buscar así cómo también recopila y envía noticias a otros usuarios y busca en nuevas fuentes. “WebMate” es un agente personal que acompaña a los usuarios de página en página proporcionándoles asistencia. Puede aprender el perfil del usuario y recopilar un diario personal, ayuda al usuario a mejorar sus búsquedas a través de la expansión de las palabras relevantes. Este agente personal acompaña al usuario cuando explora y busca información, además le proporciona ayuda “inteligente”. “WebMate” es capaz de aprender de manera incremental los intereses del usuario con actualización continua y automáticamente proporciona documentos que le interesan al usuario. También es capaz de ayudar al usuario a definir la búsqueda e incrementar la recuperación de información en documentos relevantes.

El “WebMate” aprende las categorías que son de interés para el usuario. Es decir, aprende el perfil del usuario incremental y continuamente. Para ello

es necesario que en un principio, el usuario le dé algunos ejemplos positivos - documentos de su interés - marcándolos con la etiqueta “Me agrada”, el “WebMate” no mantiene o guarda los ejemplos positivos, sino que más bien mantiene el perfil aprendido de estos ejemplos, de esta manera el sistema se adaptará a la evolución del usuario y a sus nuevos intereses. Para ello “WebMate” utiliza múltiples vectores de representación.

Hasta ahora se han explicado sistemas donde el trabajo es realizado por un único agente, en seguida se muestran trabajos que han empleado sistemas multi-agente:

Un sistema interesante es “Amalthea” (Moukas 1996), que trata de descubrir información desde varias fuentes distribuidas que interesan al usuario o usuarios, plantea, que es necesario aprender los hábitos e intereses de los usuarios y adaptarse a él siguiendo sus intereses. “Amalthea” se implementó al crear un ecosistema de agentes que evolucionan, cooperan y compiten en un ambiente de recursos limitados. Se tienen dos especies de agentes, agentes que filtran¹⁴ la información y agentes que descubren la información. Los agentes que filtran la información son los responsables de la personalización del sistema y de mantener (y adaptarse a) la ruta de intereses del usuario. Los agentes que descubren la información son responsables de mantener las fuentes de información (cuidando los cambios en ellas o descubriendo otras nuevas), de adaptarse a estas fuentes, de encontrar y obtener la información actual en la que el usuario está interesado. Se utilizan técnicas de algoritmos genéticos como las mutaciones y la cruce, así como también la especialización de los agentes de la población. “Amalthea” se inspira en la IABCm utilizando mecanismos de aprendizaje y de representación completamente distribuidos.

Un sistema multi-agente dedicado a filtrar información en noticias es “NewT” (Sheth 1994). Este sistema utiliza algoritmos genéticos para filtrar la información. Es capaz de mostrar nuevos artículos por medio de la mutación. Está compuesto por tres módulos: Un módulo interfaz a través del cuál se da la retroalimentación entre el usuario y el agente. Un módulo de aprendizaje por medio del que “NewT” aprende los intereses y perfiles del usuario. Y el módulo de filtrado de información, el cual utiliza los perfiles del usuario

¹⁴ Por el término “filtrar” información se entiende sólo considerar la información relevante para el usuario.

como entrada para encontrar los artículos que sean compatibles con dicho perfil.

La manera en que estos sistemas que utilizan agentes que colaboran entre sí se explica en (Foner 1995), en donde se plantea que las aplicaciones futuras de los agentes son las que involucran a gran número de agentes interactuando entre sí. Se espera que el agente de un usuario que actúa en su nombre se comuniquen con otros agentes de otros usuarios en cualquier lugar de la red o del ambiente. El conjunto total podría parecerse a una ecología computacional donde cada agente sólo cuenta con conocimiento local, pero en grandes unidades son capaces de auto-organizarse. El problema que se presenta entonces, es la agrupación de agentes semejantes, de modo que cada agente pueda encontrar a otro que pueda ayudarlo a través de la red. A continuación se discuten ejemplos de éstos agentes.

(Lashkari et. al. 1994) propone que varios agentes se ayuden colaborando, que los agentes experimentados ayuden a los nuevos agentes “inexpertos” en las situaciones desconocidas. Se tienen diferentes agentes que ayudan a diferentes usuarios -los agentes pueden ser basados en el conocimiento o en el comportamiento-, pero pueden cooperar para conseguir el mejor resultado para su usuario individual. Así que con el tiempo un agente aprende cuál de los agentes que cooperan con él es el más confiable, es decir, el que le proporciona sugerencias más acertadas de acuerdo con las necesidades de su usuario. El agente aprende almacenando en su memoria el par de situación-acción, este par se obtiene cuando el usuario realiza una acción y la empareja con la correspondiente situación. Para que los agentes puedan comunicarse y colaborar deben manejar el mismo lenguaje y seguir un protocolo común. Todos los agentes se registran en una “tabla de agentes” dónde cada agente tiene un identificador único que le es asignado, además todos los agentes conocen la ubicación de esta tabla. En esta tabla cada agente tiene información de cómo ser contactado, del protocolo que maneja, así como del dominio sobre el cual puede ofrecer ayuda. La comunicación entre los agentes se da a través de mensajes de solicitud y respuesta.

Otro ejemplo de un sistema en donde varios agentes colaboran es “Challenger” (Chavez et. al. 1997). Es un sistema multi-agente que asigna recursos distribuidos (y en particular asigna tiempo del procesador), los agentes que intervienen se comunican entre sí para hacer un uso más eficiente

de tales recursos. Utiliza los principios de la economía de mercado para conseguir por medio de interacciones locales entre los agentes un comportamiento global coherente. Ayuda al usuario minimizando el flujo del tiempo, es decir, haciendo que sus programas se ejecuten más rápido. “Challenger” es un sistema completamente descentralizado y distribuido donde cada agente se ejecuta localmente sobre una máquina en la red. Todos los agentes son semejantes en el sentido de tener el mismo comportamiento. El ejemplo, aunque ajeno al área de búsqueda y recuperación de información, es ilustrativo en lo referido a sociedades de agentes.

Una técnica que se relaciona con nuestra área de estudio es la Minería de Datos o *Data Mining*. *Data mining* puede definirse cómo la Extracción de Conocimientos en Grandes Bases de Datos. *Data mining* es una técnica para buscar relaciones y patrones globales escondidos en grandes bases de datos, tales como relaciones entre los datos de los pacientes de algún hospital y su diagnóstico médico (Holsheimer & Siebes 1994; Jasso 1995). Se define como "la extracción no trivial de información implícita, previamente no conocida y potencialmente útil a partir de datos" (Jasso 1995). *Data mining* utiliza la Lógica Inductiva. A diferencia de la Lógica Deductiva, en la que todos los resultados (deducciones) son lógicamente correctos, la Lógica Inductiva generaliza a partir del contenido de la BD. Extrae información de alto nivel, o conocimiento. El conocimiento que es extraído se almacena en forma de reglas. Por ejemplo, después de ver cien plantas, y notar que todas son de color verde, se puede inducir que todas las plantas en el mundo son de color verde¹⁵, aunque esto sea incorrecto, pues existe el caso de las algas rojas, la información nos es útil. El conocimiento puede ser empleado en el análisis de datos, aplicaciones de soporte de decisiones y sistemas de información ejecutiva, por ejemplo: Predecir si alguien va a pagar un préstamo, con base a datos como su salario, escolaridad, número de automóviles que tiene y el valor de su casa. Un programa puede inducir que los envíos a una bodega nunca incluyen cierto producto y notificar en el momento que suceda para corregir el error. Uno de los mayores problemas con *Data mining* es que el número de posibles relaciones es demasiado grande, y se vuelve prácticamente imposible el validar cada una de ellas: antes de hallar la relación entre un diagnóstico de hepatitis con los hábitos alimenticios del paciente, es necesario buscar la relación con un sin número de otros factores:

¹⁵ Cómo resultado se tienen reglas cómo: SI objeto = planta ENTONCES color-objeto = verde (Jasso 1995).

tipo de sangre, edad, sexo, entre otros. Otro problema es que la información muchas veces está corrompida, tiene ruido, o simplemente le faltan partes (Holsheimer & Siebes 1994; Jasso 1995). Para esto, se aplican técnicas estadísticas que ayudan a estimar la confiabilidad de las relaciones halladas. *Data mining* opera sobre grandes cantidades de datos y puesto que la finalidad original de los datos no es la de ser utilizados para extraer conocimiento, pueden contener errores, ruido y hasta contradicciones. La calidad de las decisiones que puedan tomarse con *Data mining* depende de la calidad de la información (Holsheimer & Siebes 1994), es necesario un procesamiento de los datos contenidos en la BD que los transforme y reduzca; este es un paso crítico dentro del proceso global de *Data mining*, que requiere un buen conocimiento del problema, y que, con frecuencia, marca la diferencia entre el éxito o fracaso de la minería de datos. La fase de búsqueda y recuperación de información en *Data mining* es decisiva para conseguir el éxito (Gómez 1998).

3.5 Trabajo Futuro.

Existen varias alternativas para continuar con nuestra investigación. Dado que cada agente-hormiga utiliza a la arquitectura subsumción, que no es una arquitectura cerrada y que permite incrementar las habilidades de nuestros agentes, se pueden tomar a corto plazo instrumentar conductas tales como:

- a) relacionar términos de raíz similar.
- b) relacionar datos por contexto.
- c) relacionar datos por "similitudes etimológicas".

La propuesta anterior implicaría filtrado de información que seguramente no mantendría eficiencia computacional. Sin embargo nos permitiría diseñar y resolver problemas de conflictos entre conductas, así como también nos permitiría implementar conductas complejas¹⁶.

Por ahora el sistema le muestra al usuario los resultados de la búsqueda en la BD, estos resultados incluyen información relevante así como información relacionada con ella. Si el usuario hiciera recomendaciones sobre

¹⁶ Una conducta compleja surge como resultado de la aplicación de diferentes conductas básicas (Mataric 1994, 1995; Muñoz 1997).

la dirección de la búsqueda con base en los resultados, representaría otra interesante propuesta para extender el sistema.

A mediano plazo nos interesa realizar experimentos en una plataforma de cómputo distribuida experimentando con un BD distribuida físicamente.

Y a largo plazo nos interesa realizar una implementación en paralelo de nuestra sociedad de agentes-hormiga, y poder comparar los resultados de la implementación en paralelo con los resultados que se obtuvieron en esta disertación con una implementación serial.

Capítulo 4.

Conclusiones.

La realización del presente trabajo nos ha llevado a obtener diversas conclusiones.

Primero, de los resultados obtenidos se considera que la implementación del sistema multi-agente que se propuso para solucionar el problema de búsqueda y recuperación de información en BD no es factible, la implementación del sistema mostró ser en mucho más lenta que los métodos tradicionales de búsqueda con los cuales fue comparada.

A pesar de estos resultados no deben descartarse las ventajas que ofrecen las Metodologías Basadas en el Comportamiento en la solución del problema de manera distribuida. Estas ventajas consisten en lograr una solución más robusta y tolerante a fallas. En los sistemas basados en el comportamiento la inteligencia se descompone en módulos y no se encuentra programada explícitamente en ninguno de ellos. Con el incremento de la modularidad se intenta reducir la complejidad del problema (Nwana 1996). Estos sistemas son reactivos, actúan en correspondencia a los estímulos de su ambiente. Han probado su desempeño en el mundo físico, donde se encuentran con situaciones variables e imprevistas, estos sistemas no requieren de una modelación global del mundo resultando en un bajo costo computacional al no requerir de un mapa global y del ambiente donde se desenvolverán ya que el mundo es su mejor modelo. Lo anterior nos conduce a considerar a las Metodologías Basadas en el Comportamiento en la creación de herramientas que nos ayuden a solucionar problemas complejos, que tradicionalmente han sido abordados desde un enfoque centralizado (Brooks 1986, Brooks 1990, Brooks 1991a, Brooks 1991b; Maes 1993).

Cómo se mencionó en la sección 3.3.1, la implementación del sistema multi-agente mostró problemas en la ejecución en tiempo real. Esto se debe a que el problema de búsqueda y recuperación de información es lo

suficientemente difícil que los agentes, aún cooperando, no pueden encontrar la solución en sólo unos cuantos pasos.

Se ha propuesto en el presente trabajo el diseño de entidades que sean capaces de desenvolverse en ambientes imprevisibles y complejos. Este diseño además de ser relativamente simple, ha hecho énfasis en la reducción de la carga computacional involucrada en la operación del sistema. Las razones que nos permiten sostener esto, son las siguientes: Con estas entidades -a las que llamamos agentes- formamos una "sociedad" de agentes reactivos que no requieren de un conocimiento global de su entorno, lo conocen de lo que de él perciben y actúan en consecuencia. Estas entidades tienen mínima capacidad cognoscitiva y aún con ello son capaces de encontrar solución a una tarea compleja. Se ha discutido que la solución grupal del problema requiere de un mecanismo de comunicación, que en nuestro trabajo fue instrumentado de manera indirecta, para lograr la auto-organización del sistema. Sin embargo los sistemas reactivos, cómo es el caso del sistema que implementamos, resulta estar entre los tipos de agentes más difíciles de diseñar e implementar (Jennings & Wooldridge 1998).

El problema que presentó la implementación de nuestra sociedad de agentes-hormiga, es uno de los problemas comunes a todas las aplicaciones basadas en agentes (Jennings & Wooldridge 1998), resultando que en ocasiones el usuario sacrifica velocidad por calidad de la información (Lesser et. al. 1998).

Como se ha mencionado antes nuestro sistema multi-agente se inspira en la metáfora del hormiguero ofreciendo una solución distribuida al problema (Langton et. al. 1995). Sin embargo una efectiva búsqueda distribuida no puede ser óptima en una búsqueda secuencial (Lander 1994).

Otras investigaciones enfocadas a la implementación de agentes cooperativos no resultaron óptimas en tiempo real, y proponen emplear una arquitectura paralela para hacer eficiente en tiempo la implementación de estos agentes (Clearwater et. al. 1992; Hogg & Williams 1993; Oates et. al. 1994; Decker et. al. 1995). Se considera también que cuando los problemas son lo suficientemente difíciles que requieren de una gran cantidad de pasos para solucionarse, aún si hay cooperación, pueden aprovecharse las computadoras paralelas para solucionar eficientemente estos problemas

complejos (Clearwater et. al. 1992; Ray & Thearling 1997). Resulta entonces, que nuestros agentes-hormiga tienen una ventaja importante en su diseño, el cual es inherentemente paralelizable. Cada agente actúa paralelamente sobre los elementos de su ambiente, trabajan en paralelo sin obedecer a una organización jerárquica, no existe una autoridad central dictando el comportamiento de cada individuo, haciendo con ello al sistema tolerante a fallas (Deneubourg et. al. 1992) (Resnick 1994). Esto nos permite pensar en implementar nuestro sistema en una arquitectura paralela en un futuro.

Con la implementación de nuestro sistema multi-agente se consiguió cumplir con los siguientes puntos:

- Se alcanzó el objetivo de aplicar las Metodologías Basadas en el Comportamiento en el diseño e implementación de agentes, capaces de contender con ambientes imprevisibles y complejos. El medio ambiente en el que se encuentran inmersos nuestros agentes es una BD, este ambiente es imprevisible o no determinista porque el comportamiento de cada agente depende de su percepción actual del mundo. Además se trata de un ambiente complejo porque el agente tiene que contender con información incompleta, fragmentada u oculta (Russell & Norvig 1995; Guerra 1997).
- Los agentes-hormiga consiguieron la recuperación grupal de la información contenida en la BD. Y con ello se consiguió el objetivo de estudiar a las sociedades artificiales mediante la experimentación.
- Se estudió e implementó el mecanismo de comunicación indirecta requerido para conseguir la colaboración entre los agentes-hormiga.

Nuestra propuesta está orientada a dar una solución cooperativa al problema de búsqueda y recuperación de información, y esperamos con nuestro trabajo contribuir en otras investigaciones tales como la IAD (Nwana 1996), en la implementación de agentes de información (Jennings & Wooldridge 1995), en la investigación de agentes de información distribuidos (Oates et. al. 1994), con el paradigma FA/C. Además el diseño de agentes reactivos inspirados en sociedades biológicas, puede contribuir con disciplina tales como la robótica y la vida artificial.

En esta disertación de igual modo se proporcionan medios de conceptualización, y se exhiben intentos de formalizaciones para las Metodologías Basadas en el Comportamiento. Si bien se tienen intentos de formalizaciones tales como la agenda de trabajo propuesta por Brooks para el diseño ascendente de agentes (Brooks 1991b), la heurística de mundos incrementalmente complejos de De la Mora para el diseño de los componentes esenciales de un individuo o agente (De la Mora 1997), ó la arquitectura subsumción que le permite a un agente elegir la conducta adecuada en cada momento (Brooks 1986), no es posible obtener una formalización en un sentido matemático debido a que esta Metodología se basa en la investigación en sistemas biológicos.

Un punto importante a considerar, es que aún debemos comprender y aprender más de los fenómenos biológicos que tomamos como metáforas para el diseño de nuestras "sociedades". Al implementar nuestras sociedades artificiales se intenta reproducir las reglas de comportamiento que produzcan estructuras similares a las sociedades biológicas (Deneubourg et. al. 1992; De la Mora 1997). Debemos apoyarnos de disciplinas como la etología, filosofía, biología, matemáticas, entre otras, para la implementación nuestros sistemas artificiales.

Referencias.

- Almeida 1992 ALMEIDA M.A.: Bases de Datos. McGRAW- HILL /INTERAMERICANA DE ESPAÑA, S.A., pp. 1-5, ESPAÑA.
- Blum et. al. 1998 BLUM T., KEISLAR T., WHEATON J.,WOLD E.: Writing a Web Crawler in the Java Programming Language. Documento disponible en: <http://info.webcrawler.com/mak/projects/>
- Bradshaw 1997 BRADSHAW J. M.: Introduction to Software Agents. En: *Software Agents*. AAAI Press/The MIT Press (Publisher). Documento disponible en: <http://www.cs.umbc.edu/agents/introduction/>
- Brooks 1986 BROOKS R.: A Robust Layered Control System for a Mobile Robot. En: *IEEE Journal of Robotics and Automation*. Vol RA-2, No. 1, pp. 14-23, USA.
- Brooks 1990 BROOKS R.A.: Elephants Don't Play Chess. En: *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*. Ed. Pattie Maes. MIT Press, pp. 3-15, USA.
- Brooks 1991a BROOKS R.A.: Intelligence Without Reason. En: *Proceedings of the 12th International Joint Conference on Artificial Intelligence, IJCAI 91*, pp. 569-595, Australia.

- Brooks 1991b
 BROOKS R.A.: Challenges for Complete Creature Architectures. En: *From Animals to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior*. Ed. Meyer J.A., Wilson S. MIT Press, pp. 434-443, USA.
- Carver & Lesser 1994
 CARVER N., LESSER V.: A First Step Toward the Formal Analysis of Solution Quality in Distributed Interpretation Systems. En: *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence*. Documento disponible en: <http://www.cs.siu.edu/~carver/umass/fac-analysis-umass.html>
- Clearwater et. al. 1992
 CLEARWATER S., HUBERMAN B., HOGG, T.: Cryptarithmic: Cooperative Problem Solving. En: *Computation: The Micro and the Macro View*. Editor B. Huberman. Publisher "World Scientific", pp. 33-70. Documento disponible en: <http://www.parc.xerox.com/spl/groups/dynamics/www/cryptarithmic.html>
- Colomi, et. al 1992
 COLORNI A., DORIGO M., MANIEZZO V.: Distributed Optimization by Ant Colonies. En: *Toward a Practice of Autonomous Systems. Proceedings of the First European Conference on Artificial Life*. Varela F., Bourgine P. (eds). MIT Press, 2nd. ed., pp. 134-142, USA.
- Covrigaru & Lindsay 1991
 COVRIGARU A.A., LINDSAY R.K.: Deterministic Autonomous Systems. En: *AI Magazine, Fall 1991*, AAAI Pub., pp. 110-117, USA.
- Chavez et. al. 1997
 CHAVEZ A., MOUKAS A., MAES P.: Challenger: A Multiagent System for Distributed Resource Allocation. En: *Proceedings of the International Conference on Autonomous Agents*, Marina Del Ray, California, USA. Documento disponible en <http://agents.www.media.mit.edu/groups/agents/publications/>

- Chen & Sycara 1997 CHEN L., SYCARA K.: WebMate : A Personal Agent for Browsing and Searching. The Robotics Institute Carnegie Mellon University, Pittsburgh, USA.
Documento disponible en:
<http://www.cs.cmu.edu/~softagents/webmate/ WebMate.html>
- De la Mora 1997 DE la MORA, R.: Propuesta de Representación para Sistemas Estrictamente Reactivos. Tesis de Maestría en Inteligencia Artificial. MIA-UV, MÉXICO.
- Decker et. al. 1995 DECKER K., LESSER V., PRASAD M. V. N., WAGNER T.: MACRON: An Architecture for Multi-agent Cooperative Information Gathering. En: *The Proceedings of the CIKM Workshop on Intelligent Information Agents*. Baltimore, Maryland.
Documento disponible en:
<http://dis.cs.umass.edu/research/cig.html>
- Deneubourg et. al. 1991 DENEUBOURG J. L., GOSS S., FRANKS N., SENDOVA-FRANKS A., DETRAIN C., CHRÉTIEN L.: The Dynamics of Collective Sorting Robot-Like Ants and Ant-Like Robots. En : *From Animals to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior*. Meyer J.A., Wilson S. (eds.). MIT Press, pp.356-363, USA.
- Deneubourg et. al. 1992 DENEUBOURG J.L., THERAULA G., BECKERS R.: Swarm-Made Architectures. En: *Toward a Practice of Autonomous Systems. Proceedings of the First European Conference on Artificial Life*. Varela F., Bourgine P. (eds). MIT Press, 2nd. ed., pp. 123-133, USA.

- Drogoul & Ferber 1993 DROGOUL A., FERBER J.: From Tom Thumb to the Dockers: Some Experiments with Foraging Robots. En: *From Animals to Animats 2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior*. Meyer J.A., Roitblat H., Wilson S. (eds). MIT Press, pp. 451-459, USA.
- Foner 1995 FONER L.: Clustering and Information Sharing in an Ecology of Cooperating Agents, Workshop Notes of the AAAI '95 Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments, Stanford University, California, USA.
Documento disponible en
<http://agents.www.media.mit.edu/groups/agents/publications/>
- Foster 1994 FOSTER S.: How to Compose Veronica queries.
Documento disponible en:
<gopher://gopher.scs.unr.edu/00/veronica/how-to-query-veronica%09%09%2B>
- Franklin & Graesser 1996 FRANKLIN S., GRAESSER A.: Is it an Agent, or just a Program? : A taxonomy for Autonomous Agents. Institute for Intelligent Systems, University of Memphis. USA.
- Franklin & Song 1996 FRANKLIN S., SONG H.: Sumpy: A UNIX Software Agent.
Documento disponible en:
<http://www.msci.memphis.edu/~anwara/sumpy.html>
- Gómez 1998 GÓMEZ A. J.: Inducción de Conocimiento con Incertidumbre en Bases de Datos Relacionales Borrosas. Tesis Doctoral. Universidad Politécnica de Madrid, Madrid, ESPAÑA.
Documento disponible en:
<http://www.gsi.dit.upm.es/%7eanto/tesis/html/>

- Guerra 1997 GUERRA A.: Agentes autónomos y Selección de Acción, una perspectiva basada en Redes de Comportamiento. Tesis de Maestría en Inteligencia Artificial. MIA-UV, MÉXICO.
- Hofstadter 1989 HOFSTADTER D. R.: Gödel, Escher, Bach: Una eterna trenza dorada. Usabiaga M. A. (trad.). CONACYT, pp. 366-395. MÉXICO.
- Hogg & Williams 1993 HOGG T., WILLIAMS C. P.: Graph Coloring: Solving the Really Hard Problems with Cooperative Search. En: *Proceedings, AAAI-93*. Publisher AAAI Press., pp. 231-236. Washington, D.C, USA.
Documento disponible en:
<http://www.parc.xerox.com/spl/groups/dynamics/www/graphColor.html>
- Holsheimer & Siebes 1994 HOLSHEIMER M., SIEBES A.: Data Mining: The Search for Knowledge in Databases. Reporte técnico 9406, pp. 1-78, Centro de Investigación en Informática de Holanda, Amsterdam, HOLANDA.
Documento disponible en:
<ftp://ftp.cwi.nl/pub/CWIREports/AA/CS-R9406.ps.Z>
ó en
<http://axp16.iie.org.mx/Monitor/v01n03/monitor.htm>
- Huetis 1997 HUESTIS D. L.: Robotics : Representative Projects and Publications.
Documento disponible en:
<http://www-mpl.sri.com/topics/robotics.html>
- Jasso 1995 JASSO H.: Data Mining: Extracción de Conocimientos en Grandes Bases de Datos. En: *Monitor Tecnológico de Cómputo*. Vol 1, núm 3. , Departamento de Cómputo del IIE.
Documento disponible en:
<http://axp16.iie.org.mx/Monitor/v01n03/monitor.html>

- Jennings & Wooldridge 1995 JENNINGS N. R., WOOLDRIDGE M. J. : Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review* 10 (2), pp. 115-152.
Documento disponible en:
<http://www.cs.umbc.edu/agents/introduction/>
- Jennings & Wooldridge 1998 JENNINGS N. R., WOOLDRIDGE M. J. : Applications Of Intelligent Agents. Jennings N. R., Wooldridge M. J. (Ed.), *Agent Technology Foundations, Applications, and Markets*, Springer-Verlag.
Documento disponible en:
<http://www.cs.umbc.edu/agents/introduction/>
- Lander 1994 LANDER S. E.: Distributed Search and Conflict Management Among Heterogeneous Reusable Agents. Ph.D. thesis, University of Massachusetts, Amherst, Massachusetts, USA. Also available as Technical Report 94-32, Computer Science Department, University of Massachusetts, Amherst.
Documento disponible en:
<http://dis.cs.umass.edu/research/meta-learn.html>
- Langton et. al. 1995 LANGTON C., MINAR N., BURKHART R.: The Swarm Simulation System A tool for studying complex systems. April 10, 1995.
Documento disponible en:
<http://www.santafe.edu/projects/swarm/>
- Lashkari et. al. 1994 LASHKARI Y., METRAL M., MAES P.: Collaborative Interface Agents. En: *Proceedings of AAAI '94 Conference*, Seattle, Washington, USA.
Documento disponible en
<http://agents.www.media.mit.edu/groups/agents/publications/>

- Lesser et. al. 1998 LESSER V., HORLING B., KLASSNER F., RAJA A., WAGNER S., ZHANG XQ. : A Next Generation Information Gathering Agent. Computer Science Technical Report 1998-72
 Documento disponible en:
<http://www.cs.siu.edu/~carver/umass/dsa-umass.html>
- Lieberman 1995 LIEBERMAN H.: Letizia: An Agent That Assists Web Browsing. En: *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, Montreal, CANADA.
 Documento disponible en:
<http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia.html>
- Lieberman et. al. 1997. LIEBERMAN H., VAN DYKE N., VIVACQUA A.: "Let's Browse, a Collaborative Browsing Agent," Agents World Conference, Workshop on Community Ware. USA.
 Documento disponible en:
<http://lieber.www.media.mit.edu/people/lieber/Lieberary/Lets-Browse/Lets-Browse.html>
- Maes & Kozierek 1993 MAES P., KOZIEROK R.: Learning Interface Agents. En: *Proceedings of the AAAI 93 Conference*, MIT Press, 459-465, Cambridge, MA, USA.
- Maes 1993 MAES P.: Behavior-Based Artificial Intelligence. En: *From Animals to Animats 2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior*. Meyer J.A., Roiblat H., Wilson S. (eds). MIT Press, pp. 2-10, USA.
- Maes 1994 MAES P.: Agents that Reduce Work and Information Overload. En: *Communications of the ACM*, Vol. 37, No. 7, pp. 31-40, USA.

- Moukas 1996 MOUKAS A. G.: Amalthea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem. En: *Proceedings of the Conference on Practical Application of Intelligent Agents and Multi-Agent Technology*, London, ENGLAND.
Documento disponible en:
<http://agents.www.media.mit.edu/groups/agents/publications/>
- Muñoz 1995 MUÑOZ A.: Desarrollo de una Interfaz Orientada a Objetos para un Mundo Sintético de Agentes Reactivos. Tesis de Licenciatura en Informática. Universidad Veracruzana, MÉXICO.
- Muñoz 1997 MUÑOZ A.: Estudio sobre el Comportamiento Grupal de Robots Reactivos modelando Conductas de Recolección y Anidamiento. Tesis de Maestría en Inteligencia Artificial. MIA-UV, MÉXICO.
- Newell & Simon 1976 NEWELL A., SIMON H.A.: Computer Science as Empirical Inquiry: Symbols and Search. En: *Mind Design*. Haugeland (ed.), MIT Press, pp. 35-66, USA.
- Nilsson 1995 NILSSON N.: Eye on the Prize. En: *AI Magazine*, Summer 1995, AAAI Pub., pp. 9-17, USA.
- Nwana 1996 NWANA H. S.: Software Agents: An Overview, The Knowledge Engineering Review. Vol. 11 No. 3, pp. 205-244.
Documento disponible en:
<http://www.cs.umbc.edu/agents/introduction/ao/>
- Oates et. al. 1994 OATES T., PRASAD M. V. N., LESSER V. R.: Cooperative Information Gathering: A Distributed Problem Solving Approach. Technical Report 94-66.
Documento disponible en:
<http://dis.cs.umass.edu/research/cig.html>

- Ray & Thearling 1997 RAY T. S., THEARLING K.: Evolving Parallel Computation Complex Systems. Volume 10, Number 3.
Documento disponible en:
<http://www.santafe.edu/~kurt/text/evpar/evpar.shtml>
- Resnick 1994 RESNICK M.: Turtles, termites and traffic jams: explorations in massively parallel microworlds. MIT Press, USA.
- Russell & Norvig 1995 RUSSELL S., NORVIG P.: Inteligencia Artificial: Un enfoque moderno. Prentice Hall Hispanoamericana, pp 33-55. MÉXICO.
- Sheth 1994 SHETH B.: NEWT: A Learning Approach to Personalized Information Filtering.
Documento disponible en:
<http://agents.www.media.mit.edu/groups/agents/publications/>
- Shoham 1993 SHOHAM Y.: Agent-Oriented Programming. En: *Artificial Intelligence 60. Elsevier Science*, pp. 51-92, USA.
- Steels 1994 STEELS L.: Emergent Functionality in Robotic Agents through on-line Evolution. En: *Proceedings of the Artificial Life Conferencia MIT*. Cambridge. MIT Press, USA.
- Turing 1950 TURING A.M.: Computing Machinery and Intelligence. *Mind* 59, pp. 433-460.
- Ünsal 1993 ÜNSAL C.: Self-organization in large populations of mobile robots. Ms.Sc. Thesis, Virginia Polytechnic Institute and State University, USA.
- Wilson 1998a WILSON E.O.: Ants.
Documento disponible en:
<http://www.eb.com:180/cgi-bin/g?DocF=micro/641/18.html>

- Wilson 1998b WILSON E.O.: Pheromone.
Documento disponible en:
<http://www.eb.com:180/cgi-bin/g?DocF=micro/554/14.html>
- Wilson 1998c WILSON E.O.: Social Insect
Documento disponible en:
<http://www.eb.com:180/cgi-bin/g?DocF=micro/463/80.html>
- Winston 1989 WINSTON P.H.: Inteligencia Artificial. Addison-Wesley
Iberoamericana, 3era. ed., MÉXICO.

Apéndice A.

Especificación de la clase agente-hormiga y del agente buscador de sinónimos.

En este apéndice se encuentra el código que corresponde a la definición de nuestro agente-hormiga (Programa A.1) y el código del agente buscador de sinónimos (Programa A.2). El agente-hormiga que pertenece a la clase *agente*., Mientras que el agente buscador de sinónimos pertenece a la clase *agente-busca-sinonimos* la cual es una subclase de la clase agente. Este código fue generado en el editor de Delphi 3.0.

Programa A.1. Definición de la Clase agente.

```
type
agente = class(Tmundo)
public
    constructor Create(NumeroCampo: integer; Grid : TDBGGrid; Tablas : TTable; P:
        integer; t:integer; flag : integer); {Inicializa la posición del agente }

    {MÓDULOS SENSORES}
    function SensaRecolectaDatoPreciso: boolean;
    function SensaRecolectaDatoEquivalente: boolean;
    function SensaRecolectaDatoSimilar: boolean;
    function SensaRecolectaDatoOculto: boolean;
```

continúa . . .

continuación . . .

```
function SensaRecolectaSinonimoPreciso: boolean;  
function SensaRecolectaSinonimoEquivalente: boolean;  
function SensaRecolectaSinonimoSimilar: boolean;  
function SensaRecolectaSinonimoOculto: boolean;  
function SensaDetenerExploracion: boolean;  
function SensaAvanzaRegistro: boolean;  
function SensaAvanzaCampo: boolean;  
function SensaFeromona: boolean;
```

{MÓDULOS MEDIADORES }

```
function MediadorRecolectaDatoPreciso : boolean;  
function MediadorRecolectaDatoEquivalente: boolean;  
function MediadorRecolectaDatoSimilar: boolean;  
function MediadorRecolectaDatoOculto: boolean;  
function MediadorRecolectaSinonimoPreciso : boolean;  
function MediadorRecolectaSinonimoEquivalente: boolean;  
function MediadorRecolectaSinonimoSimilar: boolean;  
function MediadorRecolectaSinonimoOculto: boolean;  
function MediadorSigueFeromona: boolean;  
function MediadorDetenerExploracion : boolean;  
function MediadorAvanzaRegistro: boolean;  
function MediadorAvanzaCampo: boolean;
```

{MÓDULOS ACTUADORES Ó CONDUCTAS BÁSICAS }

```
procedure RecolectaDatoPreciso;  
procedure RecolectaDatoEquivalente;  
procedure RecolectaDatoSimilar;  
procedure RecolectaDatoOculto;  
procedure RecolectaDatoSinonimo;  
procedure RecolectaSinonimoPreciso;  
procedure RecolectaSinonimoEquivalente;  
procedure RecolectaSinonimoSimilar;  
procedure RecolectaSinonimoOculto;  
procedure RecolectaDatoConFeromona;  
procedure DetenerExploracion;  
procedure AvanzaRegistro;  
procedure AvanzaCampo;
```

continúa. . .

continuación . . .

```
{MECANISMO DE ACTIVACIÓN DE CONDUCTAS }
```

```
procedure jerarquia;
```

```
{FUNCIONES AUXILIARES }
```

```
function EliminaEspacios(Cadena: string): String;
```

```
function BuscaCadenaOculto(TextoBaseDatos: string; TextoABuscar: string): boolean;
```

```
function BuscaSinonimoOculto(TextoABuscar: string): boolean;
```

```
procedure AgregaCampos(NCampo: integer); {Crea la lista que contiene a los campos }  
                                         {que debe explorar el agente }
```

```
procedure Posicionar;    {Posiciona al agente en el campo que le corresponde. }
```

```
{DECLARACIÓN DE VARIABLES DE INSTANCIA }
```

```
protected
```

```
GridBD : TDBGrid;      {Identifican al mundo }
```

```
Tabla : TTable;
```

```
ncampo : integer;      {Campo donde está posicionado el agente }
```

```
{VARIABLES QUE IDENTIFICAN EL LUGAR DONDE EL AGENTE DEPOSITA  
LA INFORMACIÓN RECOLECTADA }
```

```
PosEnGrid :integer;
```

```
NTabla :integer;
```

```
PosicionDeCampos: TStringList;
```

```
Minoria: integer;
```

```
TurnoTermino : integer;
```

```
end;
```

Programa A.2. Definición de la Clase agente_busca_sinonimo.

```
type
  agente_busca_sinonimo = class(agente)
  public
    constructor Create(NumeroCampo: integer; Grid : TDBGrid; Tablas : TTable; P: integer;
      t:integer);

    {SENSORES}
    function SensaAvanzaRegistro: boolean;
    function SensaRecolectaDatoPreciso: boolean;
    function SensaRecolectaDatoEquivalente: boolean;
    function SensaRecolectaDatoSimilar: boolean;
    function SensaRecolectaDatoOculto: boolean;

    {MEDIADORES}
    function MediatorRecolectaDatoPreciso : boolean;
    function MediatorRecolectaDatoEquivalente: boolean;
    function MediatorRecolectaDatoSimilar: boolean;
    function MediatorRecolectaDatoOculto: boolean;
    function MediatorAvanzaRegistro: boolean;

    {CONDUCTAS BÁSICAS}
    procedure AvanzaRegistro;
    procedure RecolectaDatoPreciso;
    procedure RecolectaDatoEquivalente;
    procedure RecolectaDatoSimilar;
    procedure RecolectaDatoOculto;

    {MECANISMO DE ACTIVACIÓN DE CONDUCTAS }
    procedure jerarquia;

    {FUNCIONES BÁSICAS}
    function PreparaTerreno(CadenaAExplorar : string):boolean; {Identifica los sinónimos}
    function AgregaSinonimos(CadenaAInvestigar: string):boolean; {Crea una lista de
      sinónimos hallados}

end;
```

Apéndice B.

Guía del usuario.

Descripción funcional.

El sistema permite explorar y recuperar información en una BD que conste de una sólo tabla o en una BD distribuida localmente. La BD puede estar en formato dBase o Paradox. Si la BD es distribuida ésta debe estar dividida en dos fragmentos.

Las características de la búsqueda son definidas por el usuario. Quien puede seleccionar las modalidades de los resultados que desea encontrar (activando o desactivando las conductas de los agentes-hormiga), puede también elegir el número de agentes-hormiga que desea enviar a buscar, puede emplear sinónimos o no en la búsqueda, así cómo también el usuario puede delimitar la búsqueda a los campos que elija.

Requerimientos mínimos del sistema.

- PC 386 o superior.
- 6 Mbytes de memoria RAM.
- Espacio en disponible en disco duro de 6 Mb.
- Windows 3.1 o superior

Instalación del sistema.

Para instalar el sistema se requiere lo siguiente:

- Contar con un equipo que cumpla con los requerimientos mínimos.
- La aplicación hormigas.exe
- La BD de sinónimos sinonimo.dbf
- Dos archivos DLL: BIVBX11.DLL (Requerido por Delphi) y AGENTDLL.DLL (Requerido por nuestro sistema, este archivo contiene la implementación de los sensores del agente).
- Para poder tener acceso a las BD, son necesarios dos discos que contienen la versión de distribución de *Borland Database Engine* (BDE).

El procedimiento a seguir para llevar a cabo la instalación es el siguiente:

1. Insertar el disco etiquetado como Disco1 de BDE y ejecute la aplicación setup.exe. Enseguida siga las instrucciones de instalación, y cuando el programa lo solicite introduzca el disco etiquetado como Disco2.
2. Crear el directorio Hormigas en la unidad de disco duro.
3. Copie en el directorio Hormigas el contenido del disco etiquetado como Disco3. El Disco3 contiene la aplicación llamada Hormigas.exe, el archivo sinonimo.dbf y los archivos DLL.

Iniciando la ejecución del sistema.

Para iniciar el programa ejecute la aplicación hormigas.exe, esta aplicación se encuentra en el directorio hormigas (Ver figura B.1).

Una vez ejecutado, aparecerá la pantalla de presentación del programa (ver figura B.2.), es necesario esperar unos segundos para que aparezca la pantalla principal del sistema (figura B.3).

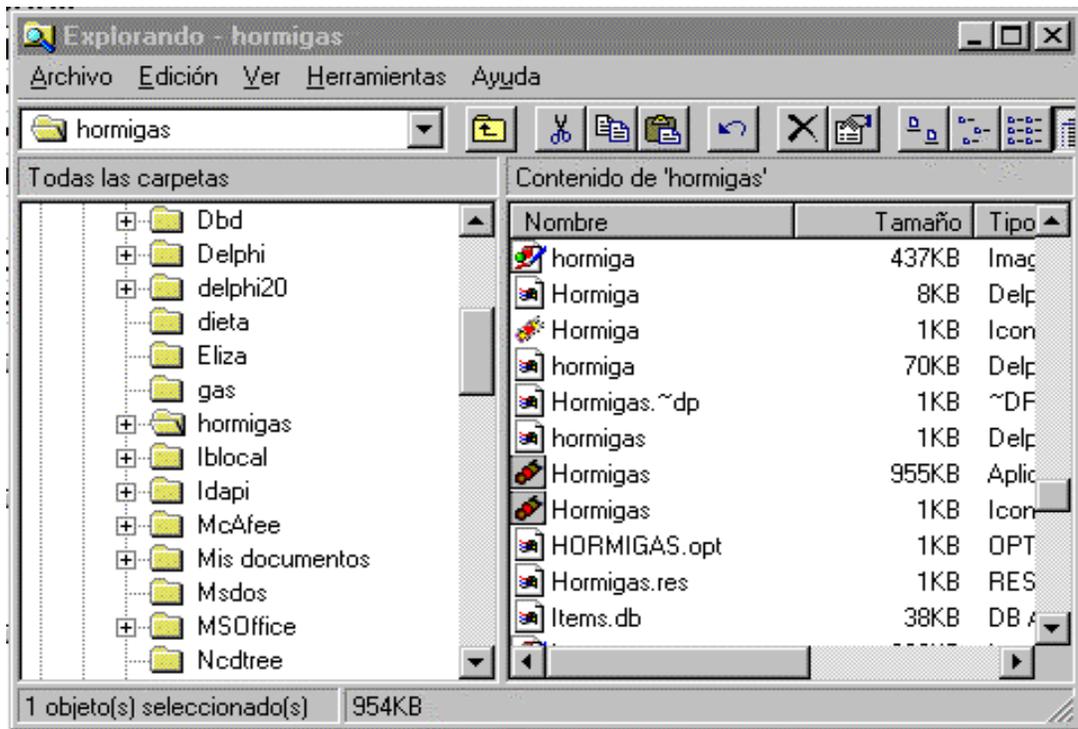


Figura B.1. Ubicación de la aplicación hormigas.exe.

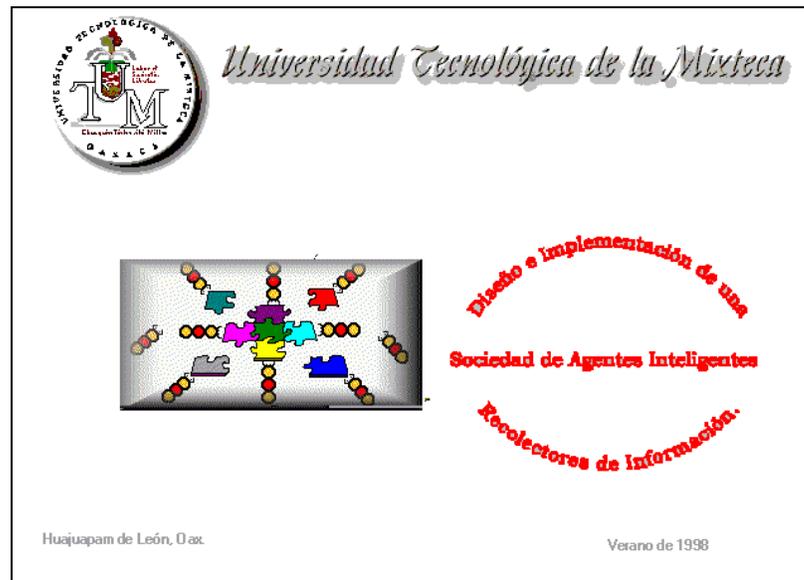


Figura B.2. Pantalla de presentación del sistema.

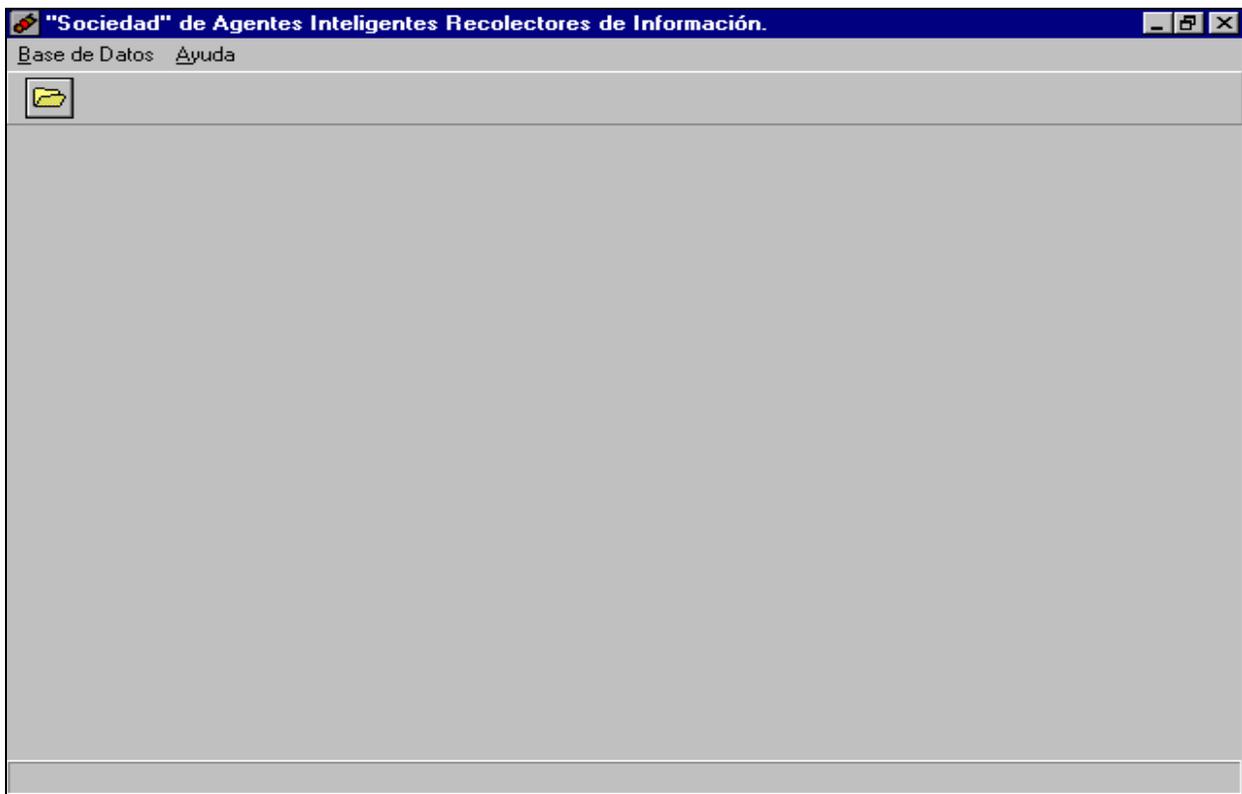


Figura B.3. Pantalla principal del sistema.

Desde esta pantalla y a través del menú se puede tener acceso a las opciones mostradas en la Figura B.4.

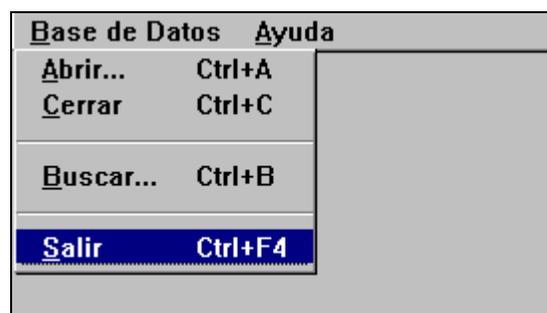


Figura B.4. Opciones del menú del sistema.

Para poder buscar la información que se desea, primero es necesario abrir la BD fragmentada o la BD única, así que en el menú sólo estarán activas en un principio las opciones de Abrir y de Salir.

Procedimiento para abrir la BD distribuida.

Para abrir la BD distribuida puede seguir cualquiera de los tres procedimientos siguientes:

1. Hacer click sobre el menú Base de Datos (o presionar Alt+B) para abrirlo. Y seleccionar la opción Abrir.
2. Presionar la combinación de teclas ctrl+A.
3. O simplemente hacer click en el icono de abrir.

En ese momento aparecerá un cuadro de diálogo solicitando el nombre y la ruta de la BD a abrir. Puesto que el sistema explora en una BD fragmentada en dos, el cuadro de diálogo aparecerá dos veces (Ver figura B.5 y figura B.6).



Figura B.5. Cuadro de diálogo que solicita el nombre de la primer BD a explorar.

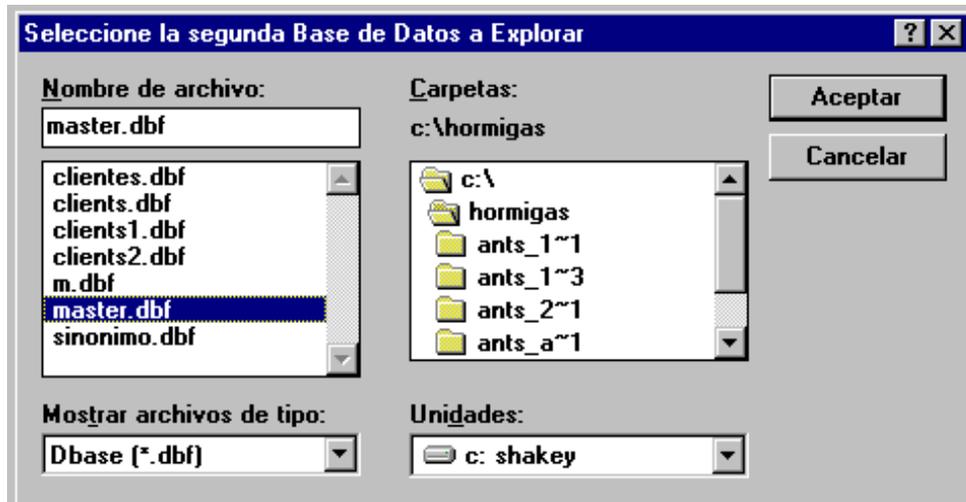


Figura B.6. Cuadro de diálogo que solicita el nombre de la segunda BD a explorar.

Si el usuario desea realizar búsquedas en una sola BD, es decir, en una BD no fragmentada, puede hacerlo. Sólo seleccione la opción de Cancelar en el segundo Cuadro de diálogo que aparece solicitando el nombre de la segunda BD que se desea abrir (figura B.6).

Una vez que ambas BD hayan sido abiertas (o que haya abierto la BD no fragmentada) son mostradas en la pantalla (figura B.7). Todo está listo para comenzar la búsqueda de la información. En el menú de Base de Datos se habilita la opción de buscar y la opción de cerrar. Si desea explorar en otras BD seleccione la opción de cerrar y repita el procedimiento anterior.

Procedimiento para cerrar la BD distribuida.

Para cerrar la BD distribuida y poder abrir otra, se abre el menú de Base de Datos y se elige la opción cerrar (que se activa al abrir una BD). O se presiona la combinación de teclas Ctrl+C.

Base de Datos Distribuida.

BD No.1

Id	last_name	first_name	acct_nbr	address_1
1	Davis	jen	1023495	100 Cranberry St
2	Jones	Arthur	2094056	10 Hunnewell St
3	Parker	Debra	1209395	74 South St
4	Sawyer	Dave	3094095	101 Oakland St
5	White	Cindy	1024034	1 Wentworth Dr

BD No.2

ID	SYMBOL	CO_NAME	EXCHANGE	CUR_PRICE	YRL_HIGH	YRL
1	SMC	STAR MOTOR CORP	NYSE	59.625	63.5	
2	TC	THORTON CORP	NYSE	27.5	28	
3	HS	HOPE SYSTEMS	NONE	29	37	
4	DHE	DHE INT'L	NYSE	8.625	9.375	
5	HGJ	HGJ CORPORATION	NYSE	42.5	43	

Figura B.7. BD abiertas.

Procedimiento para iniciar la búsqueda de información.

Abra el menú de Base de Datos y seleccione la opción de buscar. En ese momento el sistema mostrará una pantalla como la que aparece en la figura B.8. Note que es en esta parte donde puede especificar las opciones de la búsqueda, para ello haga click sobre *opciones* o presione Alt+O. Las opciones que se tienen se observan en la figura B.9.

En la parte de opciones puede seleccionarse las conductas que desea activar en los agentes-hormiga para darle mayor versatilidad a la búsqueda. Entre más conductas se activen se tendrán resultados más completos, pero también se compromete más tiempo en la búsqueda. Note que la opción de utilizar sinónimos en la búsqueda ayuda al usuario a realizar búsquedas más completas. Se tiene otra opción que delimita la búsqueda, permitiendo elegir los campos sobre los cuales buscar.

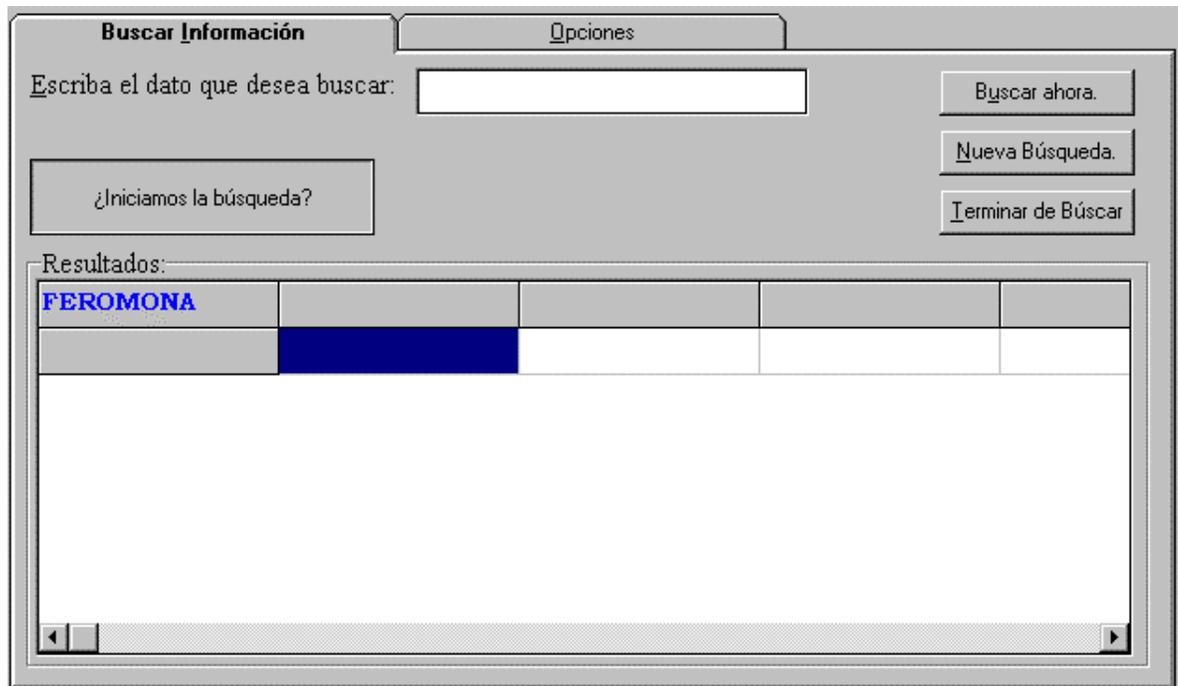


Figura B.8. Pantalla que indica que es posible iniciar la búsqueda.

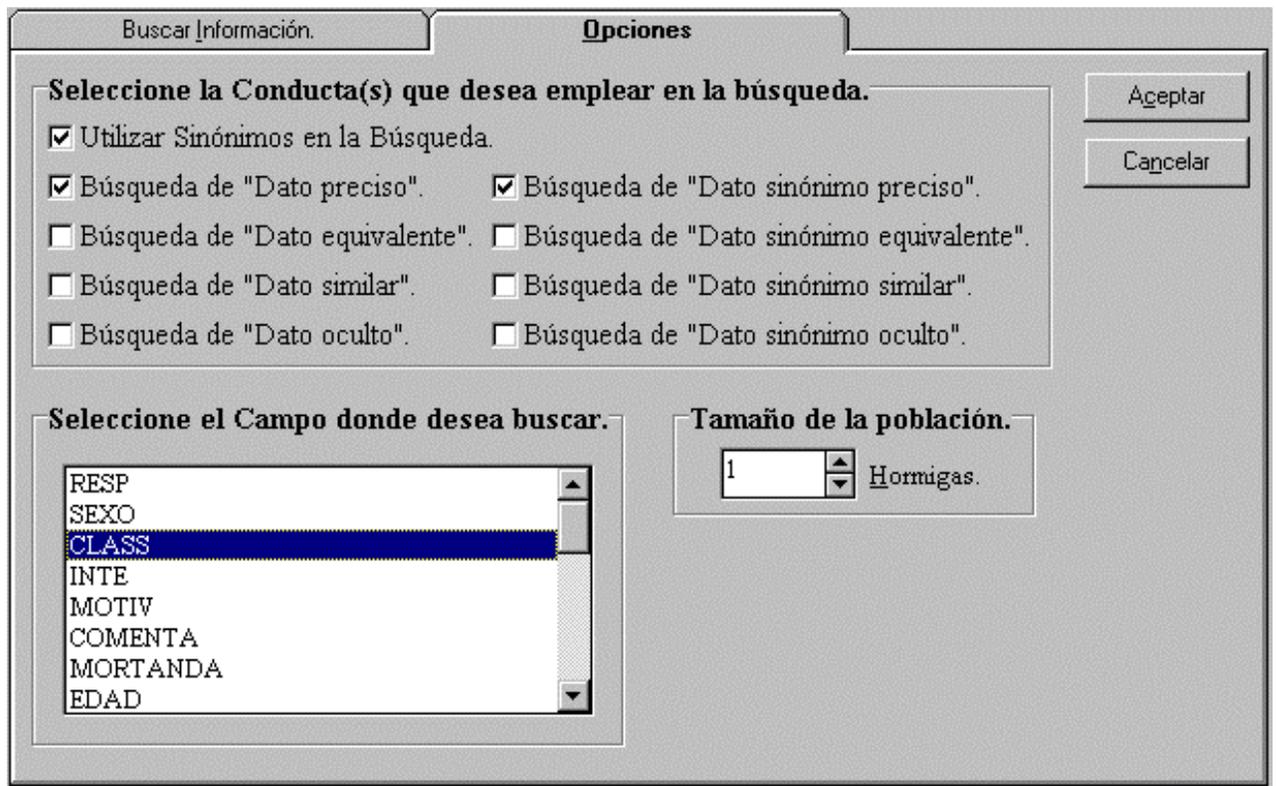


Figura B.9. Pantalla que muestra las opciones de la búsqueda.

Hay una opción que hasta ahora no se ha mencionado, esta se refiere al tamaño de la población. Esta opción permite aumentar el número de hormigas, el término hormigas se usa debido a que el sistema resuelve el problema de búsqueda y recolección de información por medio de una “Sociedad” de agentes recolectores de información, esta “sociedad” está inspirada en la metáfora de las hormigas, de ahí su relación. Esta opción está enfocada a los usuarios expertos que desean experimentar variando este parámetro. Para los usuarios inexpertos el parámetro del número de hormigas será fijado por el sistema, este parámetro depende del tamaño de las BD que se desee explorar, el número mínimo de hormigas que puede seleccionarse es igual al número de campos que hay en la BD (ó es igual a la suma del número de campos de las dos BD en caso de tratarse de una BD fragmentada)..

Para que las modificaciones sobre las opciones tengan efecto haga click en el botón de Aceptar. O haga click en Cancelar para ignorar las modificaciones.

Una vez que ya decidió los parámetros que empleará en la búsqueda diríjase a la sección de Buscar Información. Escriba el dato que desea encontrar en el cuadro de texto que tiene la leyenda “Escriba el dato que desea buscar”, y haga click en el botón de Buscar Ahora. Una vez terminada la búsqueda el sistema mostrará los resultados. Para realizar una nueva búsqueda haga click en el botón que tiene la leyenda de “Nueva Búsqueda”. El botón con la leyenda de “Terminar de buscar” oculta la sección de búsqueda y cierra la BD distribuida. Un ejemplo de los resultados obtenidos al realizar una búsqueda se muestran en la figura B.10

Buscar Información Opciones

Escriba el dato que desea buscar: [Buscar ahora.]

Información relevante encontrada.

Resultados:

FEROMONA	last_name	first_name	acct_nbr	address
1	Davis			
6	DaVis			
7	D a v i s			
11	Nelson	Roberto	40027	La casa
34	DaviSutherland			
85	d a v i s W.			

Figura B.10. Resultados obtenidos al realizar la búsqueda del dato “davis”.

Los resultados muestran en la cabecera el nombre de los campos de la BD donde se localizó la información. El encabezado Feromona, no pertenece a ninguna de las BD que integran a la BD distribuida en la que se explora, los datos que se encuentran bajo dicho encabezado no son parte de los datos recuperados en la búsqueda, más bien hacen referencia a un aspecto técnico de nuestro sistema que se utiliza para recuperar la información.

Procedimiento para salir del sistema.

Para salir del sistema haga abra el menú de Bases de Datos y elija la opción salir o presione la combinación de teclas Ctrl+F4.