

# RECUPERACIÓN DE INFORMACIÓN UTILIZANDO ECOSISTEMAS DE AGENTES INTELIGENTES

## INFORMATION RETRIEVING USING INTELLIGENT AGENTS

\***Jorge Eduardo Ortiz Triviño**

\*\***Nelson Daniel Ochoa Avendaño**

### Resumen

La inteligencia Artificial Distribuida es la ciencia encargada de estudiar conjunta e individualmente los agentes inteligentes con el fin de desarrollar nuevas soluciones a problemas de información que generalmente necesitan de una solución basada en entidades inteligentes. Los Sistemas Multiagente SMA permiten la integración con otras ciencias de la computación para desarrollar nuevas soluciones, por ejemplo con la Computación Bioinspirada. Profundizando en el tema, este trabajo presenta una investigación de la IAD para aplicarla finalmente en la implementación de un prototipo recuperador de información cuyos resultados se basan en palabras de interés del usuario, los documentos que el sistema busca se encuentran en un directorio local en formato html, la adaptación e inteligencia del sistema se hace mediante la evolución de los agentes. El sistema es diseñado bajo INGENIAS y su implementación se hace en JAVA utilizando las librería de JADE e IR.

**Palabras clave:** Inteligencia Artificial Distribuida, Agentes Inteligentes, INGENIAS, JADE,  
Ecosistema de agentes

---

\* Profesor Asistente. Departamento de Ingeniería de Sistemas e Industrial. Universidad Nacional de Colombia. Sede Bogotá.  
jeortizt@unal.edu.co

\*\* Ingeniero de Sistemas. Departamento de Ingeniería de Sistemas e Industrial. Universidad Nacional de Colombia. Sede Bogotá.ndaniel8a@yahoo.com

## **Abstract**

The Distributed Artificial Intelligence is the science responsible of studying joint and individually the intelligent agents in order to develop new solutions for information problems, which generally need a solution based on intelligent entities. The Multi-agent Systems (MAS) allow integration with other computation sciences to develop new solutions, as with the Bio-inspired Computation. To deepen in this subject, this work presents an investigation about the DAI to be applied to the implementation of a information retriever prototype which results are based on interesting works for user, the documents the system searches are found in a local directory in html format, the adaptation and intelligence of system is done through evolutions of agents. The system design is based in INGENIAS and its implementation is done in JAVA using the JADE and IR libraries.

**Keywords:** Distributed Artificial Intelligence, Intelligent Agents, INGENIAS, JADE, Ecosystem of Agents.

## **INTRODUCCION**

Este trabajo se centra en la Inteligencia Artificial Distribuida (IAD) que presenta entidades llamadas agentes para la solución de problemas que se pueden solucionar haciendo una distribución del problema, esta se hace mediante la asignación de tareas y conocimiento a cada agente, la solución del problema se logra mediante la integración de las subsoluciones presentadas por cada uno de ellos.

La IAD se puede integrar con otras áreas de la ingeniería de sistemas para formar mejores soluciones,

---

en este caso la integración se hace con los algoritmos genéticos para poder evolucionar agentes en el sistema y así dotar al sistema propuesto de inteligencia mediante la adaptabilidad. En el lenguaje de los investigadores de agentes e IAD esta integración se conoce como un ecosistema de agentes. Con la anterior idea, se diseñó e implementó un sistema prototipo para encontrar una solución al problema de la falta de efectividad de los sistemas de recuperación de información. La recuperación se hace específicamente con base en unos intereses del usuario que son plasmados mediante palabras claves y pesos, los cuáles son asignados a los agentes de filtrado para hacer las veces del usuario cuando se desea buscar información en una colección de documentos.

Para implementar el sistema propuesto, se profundizó en aspectos claves como la utilización de una metodología para el desarrollo e implantación de sistemas de IAD. En el trabajo dicha metodología se llevó a cabo mediante la investigación hecha sobre INGENIAS para el análisis y desarrollo; para la implementación del prototipo se utilizó JADE.

Este documento presenta la descripción de las diferentes etapas que se llevaron a cabo para la construcción del prototipo, en la primera parte se presenta las teorías necesarias que ayudan a tener una correcta concepción de los sistemas multiagente, en la segunda parte se presenta el problema de la recuperación de información, métodos de evaluación de estos sistemas, en la tercera parte se especifica el sistema multiagente, en la cuarta se presenta la experimentación y pruebas del prototipo, en la quinta parte se presentan las recomendaciones y conclusiones y finalmente se presentan las fuentes bibliográficas utilizadas para el desarrollo de las diferentes etapas del trabajo.

## **INTELIGENCIA ARTIFICIAL DISTRIBUIDA Y SUS AREAS DE TRABAJO**

La Inteligencia Artificial Distribuida (IAD) aparece en la década de los años ochenta como una nueva rama de la Inteligencia Artificial (IA) que tiene el fin de estudiar sistemas inteligentes formados por un

conjunto de varias entidades, éstas intentan resolver problemas en donde una conducta colectiva es más eficiente que una conducta individual.

Según Alonso [ALON 1999] la Inteligencia Artificial Distribuida es una subrama de la IA que se centra en la resolución de problemas mediante aplicación tanto de técnicas de la Inteligencia Artificial como de múltiples solucionadores de sistemas. Se involucran además mínimo dos agentes que funcionarían como solucionadores de problemas, estos agentes serían autónomos o semi-autónomos, tendrían un cierto conocimiento del problema y actuarían de forma razonable.

Los problemas básicos que estudia la IAD [BOND 1988] que son comunes a todos los sistemas son:

- Cómo formular, describir, descomponer y asignar problemas y cómo sintetizar los resultados en un grupo de agentes inteligentes.
- Cómo capacitar a los agentes para que se comuniquen e interactúen, es decir, qué tipo de lenguaje y qué protocolos de comunicación utilizarán, qué y cuándo se comunicarán, etc.
- Cómo asegurar que los agentes actúen coherentemente a la hora de tomar decisiones o de ejecutar una acción, equilibrando los efectos globales de las decisiones locales y evitando interacciones perjudiciales;
- Cómo diseñar los agentes individuales de manera que puedan representarse y razonar acerca de las acciones, los planes y el conocimiento de otros agentes, de modo que puedan coordinarse.
- Cómo reconocer y reconciliar puntos de vista diferentes e intenciones conflictivas.
- Cómo utilizar técnicas de ingeniería de software y desarrollar sistemas con IAD.

Los anteriores problemas básicos que intenta atacar la IAD se solucionan mediante diferentes teorías que se ven reflejadas en distintas áreas de trabajo, las cuales se pueden descomponer en cuatro perspectivas como se muestra en la figura 1. [IGLE 1998]

## Perspectiva de agente

Varios investigadores y grupos de investigación han definido el término de agente desde diferentes puntos de vista, esto ha influido a que en la actualidad existan diferentes definiciones de lo que es un agente. La dificultad se debe especialmente a que los agentes se pueden emplear en muchos dominios de aplicación, llevando consigo a que cada dominio asocie varios y diferentes atributos a la definición de agente. Por lo tanto es conveniente dar una corta definición de agente que no involucre las características que debe tener un agente inteligente. Un agente es un sistema que está en algún ambiente y que es capaz de tomar acciones autónomas de acuerdo al estado del ambiente para cumplir sus objetivos de diseño [WOOL 1995].

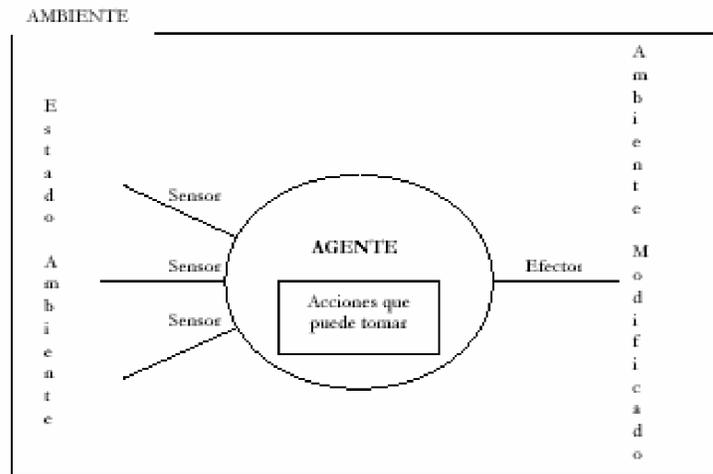
**Figura 1. Áreas de trabajo de la Inteligencia Artificial Distribuida (adaptado de IGLE 1998)**



Varios investigadores y grupos de investigación han definido el término de agente desde diferentes puntos de vista, esto ha influido a que en la actualidad existan diferentes definiciones de lo que es un agente. La dificultad se debe especialmente a que los agentes se pueden emplear en muchos dominios de aplicación, llevando consigo a que cada dominio asocie varios y diferentes atributos a la definición de agente. Por lo tanto es conveniente dar una corta definición de agente que no involucre las características que debe tener un agente inteligente. Un agente es un sistema que está en algún

ambiente y que es capaz de tomar acciones autónomas de acuerdo al estado del ambiente para cumplir sus objetivos de diseño [WOOL 1995].

**Figura 2. Esquema de la definición de agente.**



La inteligencia de un agente va asociada con el comportamiento del agente, es decir las acciones que un agente debe tomar ante un estado del entorno para cumplir sus objetivos y la capacidad del agente de interactuar con otros. En [WOOL 1995] un agente inteligente es un agente que debe tener un comportamiento autónomo y flexible, donde flexible quiere decir que el agente debe ser reactivo, proactivo y que tenga habilidades sociales.

- Autonomía: Un agente debe funcionar sin la intervención directa de otros agentes o humanos, además debe tener control de sus acciones y su estado interno [JENN 1995].
- Reactividad: Ya que un agente es capaz de percibir su entorno, éste debe responder con una acción oportuna ante un cambio ocurrido en el ambiente. [WOOL 1995].
- Pro-Actividad: Los agentes deben tener un comportamiento orientado a objetivos, por lo tanto ante cualquier estado del ambiente el agente puede tomar la iniciativa de ejecutar acciones para satisfacer sus objetivos. [WOOL 1995].

- **Habilidad Social:** Los agentes deben ser capaces de interactuar con otros agentes, aplicaciones e incluso humanos. [WOOL 1995].

## **Perspectiva de grupo**

La formulación, ejecución y coordinación de planes ha sido el camino que la IAD ha tomado para desarrollar los sistemas de agentes. Un plan en IAD se ve frecuentemente como una secuencia de acciones de un agente, de tal manera que es capaz de cambiar su entorno y lograr uno o más objetivos [LUBO 2002].

Existen dos tipos de sistemas para poder desarrollar planes en la IAD, el primero es una planificación centralizada en el cual existe un agente que formula y coordina el plan que es ejecutado por un conjunto de agentes, este tipo de sistema es conocido como Solución de Problemas Distribuidos (Distributed Problem Solving - DPS). El segundo es una planificación descentralizada en donde las tareas de formulación, ejecución y coordinación de los planes son repartidas a un conjunto de agentes, este tipo de sistema es conocido como Sistemas Multiagente (MultiAgent System - MAS) [ALON 1999].

Existen diferentes métodos de planeación, entre ellas están:

*Planeación para múltiples agentes:* La planeación para múltiples agentes puede ser considerada como un problema distribuido que los agentes tienen que resolver. [LUBO 2002].

*Planeación distribuida:* En la planeación distribuida, las tareas y resultados de esta, pueden ser fuentes de distribución en un MAS, es decir el proceso de planeación es distribuido entre muchos agentes. El resultado de este proceso de planeación es un plan que es ejecutado por varios agentes. [LUBO 2002]

## *Coordinación*

La coordinación en los sistemas de inteligencia artificial distribuida (DPS y MAS) es definida por Jennings [JENN 1996] como el proceso de razonamiento de un agente sobre sus acciones locales y anticipadamente las acciones de otros agentes, para tratar de asegurar actos coherentes en la comunidad de agentes, esta debe asegurar que las actividades de un problema global sean incluidas mínimo en un agente, que la solución de las actividades desarrolladas sea integrada en una solución global y que al desarrollar las actividades se logren los objetivos en un entorno con recursos limitados. [BENG 2001]

Existen diferentes métodos para tener coordinación en la IAD, entre estos están:

*Coordinación Organizacional:* Es un método de coordinación entre agentes que utiliza una estructura organizacional (jerárquica, plana) de la comunidad de agentes. Esta es la técnica de coordinación más simple, se asemeja a una arquitectura maestro/esclavo, ya que el agente maestro asigna tareas y recursos a los agentes esclavos. El agente maestro puede crear planes y asignar tareas a los agentes de la comunidad de modo que asegure la coherencia del sistema. [GREE 1997]

*Coordinación por contratos - El protocolo Contract Net:* Es la técnica de coordinación más conocida para la asignación de tareas y recursos entre agentes; determina la estructura organizacional “Protocolo de red de Contratos” (CNP). El problema de asignar los subproblemas se soluciona mediante un mecanismo de contratación que consiste en el anuncio de un contrato por parte del agente gerente (contratante), la creación de ofertas por los agentes contratistas en respuesta al anuncio, y la evaluación de las ofertas propuestas por el contratante que lleva a otorgar un contrato del subproblema al contratista(s) con la oferta(s) más apropiada(s). [GREE 1997]

*Coordinación por intercambio de información Meta-Nivel:* El intercambio de información Meta-nivel

implica que los agentes envíen a los otros agentes del grupo información de control por niveles sobre sus objetivos y creencias [BENG 2001].

### ***Comunicación***

La comunicación entre agentes es importante para mejorar la coordinación y coherencia global del sistema, ya que permite sincronizar las acciones, el conocimiento, y resolver conflictos en la resolución de un plan [IGLE 1998] [CORC 1999]. En los sistemas DPS se utiliza una comunicación preestablecida en la que existen señales e interpretaciones fijas mientras que en los sistemas MA se puede optar por no comunicarse para no interferir en las creencias y actividades de otros agentes hasta una comunicación de alto nivel. [ALON 1999].

### **Perspectivas Particulares**

#### ***Ecosistema de Agentes***

Los ecosistemas de agentes se crearon para proveer a los sistemas multiagente de adaptabilidad ante un cambio en el entorno que lo rodea. Justamente la inteligencia que puede tener un Sistema Multiagente se basa en esta idea novedosa que integra los algoritmos genéticos que hace parte de la Computación Bioinspirada con los agentes inteligentes que son estudiados por la IAD.

Un ecosistema de agentes consiste en una población de agentes inicial, cuyo conocimiento se basa en un fenotipo que contiene datos que no evolucionan al crear una nueva descendencia y un genotipo que mantiene datos que evolucionan al crear descendencia. Dado el conjunto inicial de agentes se escogen aquellos que hayan presentado mejores resultados al sistema para crear una descendencia de nuevos agentes. Los agentes cuyo ciclo vital está acabando son eliminados del sistema multiagente. El fitness del agente representa la vitalidad del agente.

En el campo de los agentes de información se han hecho investigaciones para crear ecosistemas de agentes destinados a la filtración. Estas proponen la asignación de un conjunto de parejas de palabras clave y pesos al genotipo del agente. La evolución de nuevos agentes se hace mediante algoritmos genéticos.

### ***Recuperación Vectorial***

En este modelo el documento se representa como una composición de vectores en los que se muestran las apariciones de cada término en el documento. A los vectores se le dan pesos que representan la importancia de los términos en el documento y en la colección entera de documentos. Cuando una palabra aparece mucho en un documento, se supone que es importante en ese documento aunque si aparece en muchos documentos el término no es útil para distinguir ningún documento del resto de la colección.

La recuperación de información se lleva a cabo calculando la similitud existente entre texto de búsqueda y los documentos que tengan los términos en el texto de búsqueda. Una técnica de este modelo es la recuperación inversa, en ésta el indexamiento consiste en la representación de palabras de los documentos como vectores de pesos. Las palabras claves son ponderadas en el documento produciendo una medida  $tfidf$  Frequency Times Inverse Document Frequency, la ponderación se representa mediante la siguiente fórmula:

$$W_i = \Pi_c \cdot T_f \cdot idf_k \quad (1-0)$$

donde  $T_f$  es la frecuencia de la palabra clave en el documento actual (frecuencia del término),  $c$   $P$  es una constante que depende de la ubicación de la palabra clave en el documento, por ejemplo si la palabra esta en el cuerpo del documento su valor es de uno, pero si se encuentra en el título del documento su valor será de dos. La expresión  $k idf$  se define formalmente como:

$$idf_k = \log\left(\frac{N}{df_k}\right) \quad (1-1)$$

donde  $N$  es el número total de documentos que han sido recuperados y  $k df$  es la frecuencia del término  $k$ -ésimo en el documento. De esta manera la expresión  $k idf$  es la frecuencia de la palabra clave en la colección entera de documentos (conjunto de todos los vectores de pesos de palabras claves).

## **PROBLEMA DE LA RECUPERACION DE INFORMACION**

El problema de los sistemas de recuperación radica en la efectividad de dichos sistemas, es decir en la relevancia de los documentos para el usuario que hace la consulta. Este problema se observa claramente cuando los sistemas de recuperación recuperan muchos documentos de los cuales sólo una pequeña cantidad son relevantes, también cuando los documentos más relevantes no aparecen necesariamente al principio de la búsqueda y cuando no se satisfacen los intereses del usuario. El problema de los sistemas de recuperación se convierte entonces en encontrar información pertinente, precisa y en el momento oportuno en una gran colección de documentos heterogéneos, [MART 2002] por este motivo los sistemas de recuperación de información se encuentran en cambio para atacar el problema de la efectividad mediante el desarrollo de nuevos métodos de búsqueda, sistemas que tratan de agregar información semántica a los documentos y nuevas formas de indexación.

### **Problema de la relevancia**

Extendiendo la definición a los documentos se puede decir entonces que un documento recuperado es relevante cuando posee alguna importancia para el usuario según sus intereses y necesidades de información [MART 2002], esto presenta inconvenientes ya que todos los usuarios no tienen el mismo

nivel de conocimiento y los mismos intereses, se destacan entonces los siguientes problemas:

- Un documento puede ser relevante o no por dos personas distintas según el nivel de conocimiento y las necesidades de información. Puede suceder además que el documento sea relevante para una persona en un instante de tiempo determinado, y en un futuro ya no serlo.
- Es difícil definir anticipadamente criterios para determinar cuándo un documento es relevante debido a la subjetividad de los usuarios.

Hay otras definiciones que dejan de lado el carácter subjetivo del conocimiento de los usuarios, se tiene a Lancaster [LANC 1993] el cual indica la relevancia como una relación entre un documento y una petición de información hecha por un usuario, es decir un documento se puede juzgar como relevante o no si hace referencia a la petición de búsqueda.

Salton [SALT 1983] determina los documentos relevantes como un subconjunto pertinente de la colección entera de documentos (pertinente que satisfacen las necesidades de información de un usuario). En esta definición se nombra un subconjunto pertinente de documentos, es decir que los documentos contenidos en los resultados deben proporcionar al usuario información que satisfaga sus necesidad de búsqueda.

### **Medidas basadas en la relevancia.**

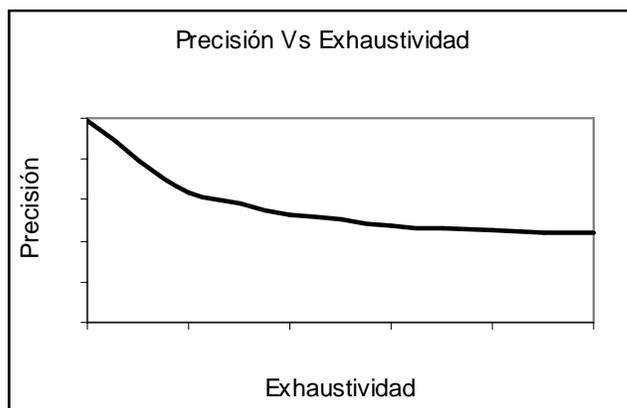
En este trabajo se da importancia a las medidas de la efectividad de la recuperación de información que se basan en la relevancia de los documentos recuperados. Se han propuesto varias medidas para evaluar la efectividad de la recuperación, a continuación se verán dichas medidas, señalando principalmente a la exhaustividad.

Durante un proceso de recuperación de información al usuario se le presentarán un cierto número de documentos, de los anteriores documentos se obtienen dos subconjuntos cuyo contenido del primero son los documentos relevantes para el usuario (docRcc-rel), y el otro esta formado por los documentos no relevantes (docRec-norel). De la misma manera al usuario no se le presentará un conjunto de documentos que serán relevantes (doc-rel) y otro que serán no relevantes (doc-norel) [MART 2002]. De los anteriores conjuntos se pueden obtener las medidas de relevancia (precisión y exhaustividad), a continuación se explicará como se pueden obtener:

- **Precisión:** Mide el porcentaje de documentos recuperados que son relevantes con el tema de interés del usuario, se calcula mediante el cociente del total de documentos relevantes recuperados (docRec-rel) entre el total de documentos recuperados (docRec-rel + docRec-norel).
- **Exhaustividad:** Es el número de documentos relevantes recuperados (docRec-rel) dividido en el número de documentos totales relevantes en la colección total (docRec-rel + doc-rel). El calculo del denominador para pequeñas colecciones puede ser hallado mediante un proceso de revisión de los documentos, si la colección es grande se debe hacer un muestreo aleatorio de los documentos de la colección y determinar de esta manera dicha cantidad [MART 2002].

Según Rijsbergen [RIJS 1999] los valores de precisión y exhaustividad tienden a relacionarse inversamente como se muestra en la figura 3, y los valores de estas cantidades deben estar entre 0 y 1, aunque en algunas ocasiones se puede representar como un porcentaje.

**Figura 3. Relación inversa entre precisión y exhaustividad.**



Se han desarrollado técnicas para evaluar simultáneamente la exhaustividad y la precisión mediante puntos de coordenadas (exhaustividad-precisión) con el fin de obtener gráficos que sigan un comportamiento algo parecido al de la figura 3.

En Salton [SALT 1983] se propone un cálculo de los valores de exhaustividad y precisión (pareja de valores Exhaustividad-Precisión ó E-P) sobre una muestra de documentos de la colección total. Durante el cálculo se obtiene una muestra pequeña de documentos de la colección, la cual está constituida por una cantidad  $N$  de documentos que se recuperan al hacer una búsqueda  $Q$ , de estos una cantidad  $R$  serán relevante y otra cantidad serán no relevantes, de la misma manera el número total de documentos relevantes en la colección de documentos será conocido.

### **Medidas alternativas**

Martínez [MART 2002] presenta otra serie de medidas denominadas medidas alternativas o medidas de valor simple las cuales se determinaron a partir de técnicas probabilísticas. Las medidas se interpretan de manera diferente, suponiendo que se hace una búsqueda cuyos resultados son ideales, es decir se recuperan 20 documentos de los cuales todos son relevantes, se obtendría entonces una exhaustividad

de uno y una precisión de uno. Los valores para las medidas vistas anteriormente serían:

- Bolko = 2 (máximo)
- Meadow = 1 (máximo)
- Heine = 0 (mínimo)
- Vickery = 0 (mínimo)

Los resultados anteriores indican que mientras mayor sea el resultado para las medidas de Bolko (se acerca a dos) y Meadow (se acerca a uno) la búsqueda será mejor, por el contrario para las medidas de Heine y Vickery mientras menor sean los resultados la búsqueda será mejor (se acercan a cero) [MART 2002].

## **HIPÓTESIS DE LA INVESTIGACIÓN**

La hipótesis de partida de la hipótesis se basa en la elaboración de un sistema de recuperación de información conformado por un ecosistema de agentes, el cuál tiene por objetivo recuperar y filtrar los documentos requeridos por un usuario.

Los agentes de filtrado tendrán un conocimiento previo de los intereses del usuario de tal manera que cada palabra tenga un peso asociado con el fin de representar más apropiadamente lo que un usuario desearía encontrar. Además los agentes pueden evolucionar con el fin de adaptarse a los cambios de interés del usuario con ayuda de las calificaciones proporcionadas.

Con el uso de los agentes de filtrado se esperaría entonces que se disminuyan los resultados presentados al usuario de tal manera que sean pocos los documentos no relevantes presentados (aumentaría precisión para los diferentes valores de exhaustividad). Por medio de la filtración también

se esperaría que algunos de los documentos relevantes no sean presentados, por lo que se estaría disminuyendo la exhaustividad del sistema y así la calidad del mismo.

También se buscaría encontrar un nivel de confianza intermedio que presente mejores medidas de evaluación, es decir un valor que permita identificar que el nivel de filtrado en los agentes es más apropiado que otros valores con el fin de aumentar la eficiencia del sistema. Lo anterior implica que se pueden obtener mejores valores de precisión para los diferentes once valores de exhaustividad y diferentes niveles de confianza, así mismo las medidas alternativas presentarían mejores o peores resultados dependiendo el nivel de confianza usado en el sistema.

Con la evolución de los agentes de filtrado se espera entonces que el sistema se adapte a los intereses del usuario, aunque se puede esperar que las calificaciones positivas hechas por el usuario a los agentes de filtrado afecten en cierta medida el proceso de filtrado de los documentos y de esta manera aumentar el número de documentos no relevantes presentados.

El problema de la relevancia entonces se ataca con los agentes de filtrado y la evolución de los mismos, los resultados del sistema en el capítulo de pruebas y resultados mostrarán si el sistema propuesto ayuda a mejorar o empeorar el problema de la efectividad, con la hipótesis planteada se esperaría que se mejore dicho problema más no se solucionaría completamente.

## **SISTEMA DESARROLLADO**

El prototipo implementado tiene como objetivo principal recuperar información que sea más acorde con los intereses del usuario. Para lograr dicho objetivo el prototipo tiene dos características fundamentales:

1. La recuperación y filtrado de recuperación se hace con base a unos intereses que el usuario plasma en palabras claves con sus respectivos pesos.
2. El prototipo puede evolucionar con los intereses del usuario mediante el desarrollo de un ecosistema de agentes, la evolución se produce gracias a las calificaciones hechas por el usuario de los documentos recuperados por cada agente, así actuará como un ente calificador de los agentes y premiará (buena calificación) a aquellos que presenten resultado que los satisfaga y castigará (mala calificación) a los agentes que presenten malos resultados, o resultados que no sean de su interés. El diseño del prototipo se hizo bajo la metodología para desarrollar SMA de INGENIAS, el desarrollo o implementación se logró mediante la utilización de JAVA y el API de construcción de SMA propuesto por JADE.

## **Arquitectura**

El prototipo de recuperación de información desarrollado en este trabajo de investigación se compone de:

- Agente Facilitador de interfaz (AI): se encarga de presentar una interfaz agradable al usuario, acá se presentan los documentos recuperados y además se hace la validación del usuario en la base de datos de información del programa.
- Agente manejador de agentes de filtrado (AFContainer): tiene como función recuperar los agentes y genotipo correspondiente a cada uno de la base de datos y hacer la evolución del sistema.
- Agentes de filtrado (AF): se encargan de solicitar a los agentes de recuperación documentos que se filtran con base a las palabras claves y sus pesos que se definen en el genotipo de cada agente de filtrado..
- Agente de recuperación (AR): se encarga de indexar los documentos encontrados en el directorio de documentos html guardados y recuperar los documentos que sean requeridos por los agentes

de filtrado.

- Base de datos de información del programa (BD Información): es el lugar en donde se almacenan los usuarios, los agentes de filtrado, las palabras claves y pesos.
- Directorio de documentos: es un directorio que contendrá los documentos en formato html donde los agentes de recuperación buscarán la información.

En el sistema cada usuario tendrá un agente facilitador de interfaz, un agente manejador de agente de filtrado, un agente de recuperación y varios agentes de filtrado que se definen en la base de datos.

*Planeación en el prototipo:* En el prototipo diseñado la planeación es distribuida, el plan se distribuye y es ejecutado por los agentes del sistema de tal manera que estos actúan tras el plan distribuido diseñado previamente, tanto la planeación como la ejecución de los planes se hace de manera distribuida.

*Coordinación en el prototipo:* Dentro del sistema se tienen diferentes tipos de coordinación con el fin de conocer que decisiones y acciones ejecutan o ejecutaron los agentes del sistema.

- Coordinación por intercambio de información Meta-Nivel: Se intercambian creencias que son obtenidas del ambiente. Este tipo de coordinación aparece en las siguientes interacciones:
  - Agente facilitador de interfaz y agente manejador de agentes de filtrado.
  - Agente facilitador de interfaz y agente de recuperación.
  - Agente de filtrado y agente facilitador de interfaz.
- Coordinación por contratos: Se coordina la prestación del servicio de recuperación por parte del agente de descubrimiento de información y los agentes de filtrado mediante el protocolo Contract Net Protocol.
- Coordinación organizacional: Hay una jerarquía entre los agentes de filtrado que actúa como un dios que puede asignar modificar y eliminar a los agentes de filtrado.

*Comunicación en el prototipo:* La comunicación entre los agentes del prototipo se hace mediante paso de mensajes de peticiones, respuestas e información, cada agente que envía un mensaje debe saber quien será el receptor para que exista una comunicación eficaz.

*Conocimiento en el prototipo:* El conocimiento de la solución global y del problema global en cada agente es incompleto y parcial, pero el conocimiento sobre la sub-solución y el sub-problema asignado en cada agente es completo. Una solución final se obtiene mediante la cooperación de los resultados propuestos por los diferentes agentes del prototipo.

*Control en el prototipo:* El control es descentralizado en cada agente que compone el sistema, cada uno se encarga de tomar sus decisiones y ejecutar las acciones según lo que perciban del entorno o lo que determinen para alcanzar sus objetivos.

### **Características del Agente facilitador de interfaz**

- Es semiautónomo ya que su comportamiento (planeación y ejecución de tareas) no es inicializado automáticamente, es necesario que el agente lea del entorno una acción del usuario con el fin de inicializarse. El control de sus acciones y el estado interno es hecho por el agente autónomamente.
- Tiene comportamiento reactivo ya que cuando se presenta un suceso en el ambiente es capaz de tomar una decisión y ejecutar una acción de acuerdo a la percepción tomada del ambiente.
- Interactúa con los agentes: manejador de agentes de filtrado y de recuperación, también interacciona con el usuario al recibir información de él y al mostrar los resultados del sistema de agentes. Además se hace interacción con la base de datos.
- Pertenece a una arquitectura reactiva por el método de planeación percepción-acción.

- El agente de interfaz persigue los siguientes objetivos:
  - Inicializar el sistema de recuperación.
  - Recibir y ofrecer información al usuario a través de la interfaz gráfica.

### **Características del Agente manejador de agentes de filtrado**

El agente manejador de agentes de filtrado permite que la recuperación de documentos se adapte al perfil del usuario. Para lograr el anterior objetivo debe hacer una evolución de los agentes de filtrado. Cada usuario dispone de un solo agente manejador de agentes de filtrado.

La evolución es controlada por el fitness global del sistema (promedio del fitness de todos los agentes de filtrado) y el fitness individual de cada agente. El número de agentes que se permiten para producir descendencia se relaciona linealmente con el número de agentes que serán eliminados. La tasa de evolución del sistema se afecta por el fitness global del sistema de tal manera que si el fitness global del sistema disminuye la tasa de evolución aumenta, pero si el fitness global aumenta la tasa de evolución se mantiene constante. Solamente se permite que haya descendencia de los agentes que tengan un buen fitness.

Los nuevos agentes son creados por cruce, dados 2 agentes se retornan 2 nuevos agentes que heredan algunas palabras claves del genotipo de cada uno de los agentes padres, la figura 10 muestra como esta constituido el genotipo del agente.

**Figura 4. Genotipo del agente de filtrado**

Palabra Interés 1	Palabra Interés 2	Palabra Interés 3	Palabra Interés 4	....	Palabra Interés n
Peso	Peso	Peso	Peso	....	Peso

Dados dos genotipos G1 y G2 de los agentes padres el operador de cruce se define como: G1 XOR G2, G3 y G4

Las palabras claves que se heredan de los agentes padres se escogen entre dos puntos del genotipo del agente.

Los puntos se obtienen de la siguiente manera:

$$P1 = \text{rand}(1, \text{sizeof}(G)-2)$$

$$P2 = \text{rand}(p1, \text{sizeof}(G)-1)$$

Los nuevos genotipos, G3 y G4 heredan una parte de las palabras claves de G1 y G2. Si i es una palabra clave de cualquier genotipo se obtiene G3 y G4 de la siguiente manera:

$$G_3 = \begin{cases} G_1 & 0 < i < p_1 \text{ y } p_2 < i \leq \text{sizeof}(G_1 - 1) \\ G_2 & p_1 \leq i \leq p_2 \end{cases}$$

$$G_4 = \begin{cases} G_1 & p_1 \leq i \leq p_2 \\ G_2 & 0 < i < p_1 \text{ y } p_2 < i \leq \text{sizeof}(G_2 - 1) \end{cases}$$

Las características del agente manejador de agentes de filtrado (AF) son:

- Es semiautónomo ya que su funcionamiento comienza cuando el agente facilitador de interfaz envía conocimiento sobre el identificador de usuario. El control y ejecución de sus acciones junto con el manejo de su estado interno se hacen autónomamente.
- Es reactivo ya que ante percepciones del ambiente este puede tomar una acción.
- Es proactivo ya que sin importar el estado del ambiente siempre está verificando si se debe hacer evolución de los agentes de filtrado.
- Interactúa con los agentes: facilitador de interfaz y agentes AF, no se hace interacción con el usuario. Además se hace interacción con la base de datos.
- Pertenece a una arquitectura reactiva por el método de planeación percepción acción.
- El agente manejador de agentes de filtrado persigue el siguiente objetivo:
  - Hacer la evolución del sistema.

### **Características del Agente de filtrado**

El agente de filtrado permite filtrar información que es recuperada por los agentes de recuperación. En el sistema se pueden crear varios agentes de filtrado para un solo usuario, todos funcionan de la misma manera pero las consultas que ellos hacen a los AR difieren en las palabras claves que cada uno tiene como genotipo.

Cada agente de filtrado determina el nivel de confianza de cada documento que es recuperado por el AR, si el nivel de confianza es el apropiado, el documento será propuesto al usuario, de lo contrario será rechazado.

El nivel de confianza se calcula con base a una puntuación (score que representa que tan relevantes son las palabras de búsqueda al documento que el AR transmite al AF) y el fitness del AF. El score de

pesos dado por el agente de recuperación es comparado con los pesos de las palabras claves del genotipo, se calcula una medida de similitud entre los dos vectores de pesos, la medida de similitud es multiplicada por el fitness del agente de filtrado que propuso el documento y finalmente se obtiene el nivel de confianza. Este nivel de confianza indica la seguridad del agente sobre el documento, solo los documentos cuyo nivel de confianza sea superior a un nivel de confianza mínimo permitido serán presentados al usuario.

Las características del agente de filtrado (AF) son:

- No es autónomo en su funcionamiento ya que la creación y eliminación se hace por medio del agente manejador de agentes de filtrado. El control de sus acciones es autónomo, el control de su estado interno es no autónomo ya que el agente manejador de agentes de filtrado modifica su fitness cuando se recibe una calificación del usuario.
- Es reactivo ya que ante percepciones del ambiente este puede tomar una acción.
- Es proactivo ya que sin importar el estado del ambiente siempre está verificando si hay propuestas de los agentes de recuperación para obtener un servicio de recuperación.
- Pertenece a una arquitectura reactiva por el método de planeación percepciónacción.
- El agente de filtrado persigue los siguientes objetivos:
  - Filtrar los documentos presentados por los agentes de recuperación.
  - Recomendar al usuario únicamente los documentos con buen nivel de confianza.

### **Características del Agente de recuperación**

El agente de recuperación se encarga de indexar los documentos y recuperar los documentos que son relevantes ante la pregunta de un AF. Los resultados se basan en el nombre del documento junto con un score que representa los pesos de las palabras que se estaban buscando.

Las características del agente de recuperación (AR) son:

- Es semiautónomo en su funcionamiento ya que el proceso de indexación solo puede empezar hasta que el agente facilitador de interfaz envíe información sobre la ruta de los documentos html. El control de sus acciones y de su estado interno se hace de manera autónoma.
- Es reactivo ya que ante las percepciones conocidas del agente sobre el ambiente puede tomar una acción o una secuencia de acciones a seguir.
- Es proactivo ya que sin importar el estado del ambiente siempre está proponiendo el servicio de recuperación de información a los agentes que lo necesiten.
- Interactúa con los agentes: facilitador de interfaz y agentes AF y de recuperación, no interactúa con el usuario ni con aplicaciones externas.
- Pertenece a una arquitectura reactiva por el método de planeación percepciónacción.
- El agente de recuperación persigue los siguientes objetivos:
  - Indexar los documentos que se encuentran en la ruta definida por el usuario.
  - Recuperar los documentos requeridos por los AF con base a las palabras de interés del usuario.

## **PRUEBAS DEL SISTEMA**

La efectividad de un sistema de recuperación de información se determina por los resultados comparativos entre documentos recuperados y documentos relevantes para una consulta específica, como ya se vio anteriormente la evaluación de la efectividad se hace mediante las formulas de precisión (P) y exhaustividad (E), especialmente como se indicó en esa sección siguiendo a Salton [SALT 1983].

La evaluación del sistema de recuperación se realizó utilizando una colección de prueba de 800

documentos html y un conjunto de veinte consultas, la tabla 1 muestra las peticiones de información que se hicieron. Los documentos presentados por el sistema de cada una de las preguntas se clasificaron en relevantes y no relevantes.

**Tabla 1. Preguntas al sistema**

Prueba	Consulta	Prueba	Consulta
1	Contaminación	11	Lenguaje aplicaciones Web PHP
2	Contaminación atmosférica	12	Anemia síntomas tratamiento
3	Agente recuperación	13	Tren alta velocidad
4	Agente filtrado	14	Propiedades nutricionales tomate
5	Agujero Negro	15	Enlace químico covalente
6	Sistema digestivo intestino	16	Función procesador computador
7	Sistema óseo huesos	17	Movimiento circular uniforme
8	Huracanes tormenta tropical	18	Patinaje velocidad
9	Literatura griega representantes	19	Campeones mundiales automovilismo
10	Lenguaje programación	20	Principales desiertos Asia

Mediante la evaluación de la relevancia se procedió a calcular las parejas exhaustividad-precisión (E-P) para cada una de las veinte consultas y luego a calcular la precisión para los once valores de exhaustividad obteniendo así los datos interpolados y sus correspondientes gráficas junto con otra gráfica de tendencia exponencial interpolada obtenida mediante la función de Microsoft Excel de agregar línea de tendencia.

Terminadas las veinte pruebas hechas al sistema se calculó la media para los once valores de exhaustividad obteniendo así valores para el sistema en general, los cuales se compararán con el modelo vectorial de recuperación. De las veinte pruebas también se obtienen las medidas alternativas vistas anteriormente para determinar un nivel de confianza adecuado para mejorar la efectividad del sistema con base al promedio de las pruebas hechas.

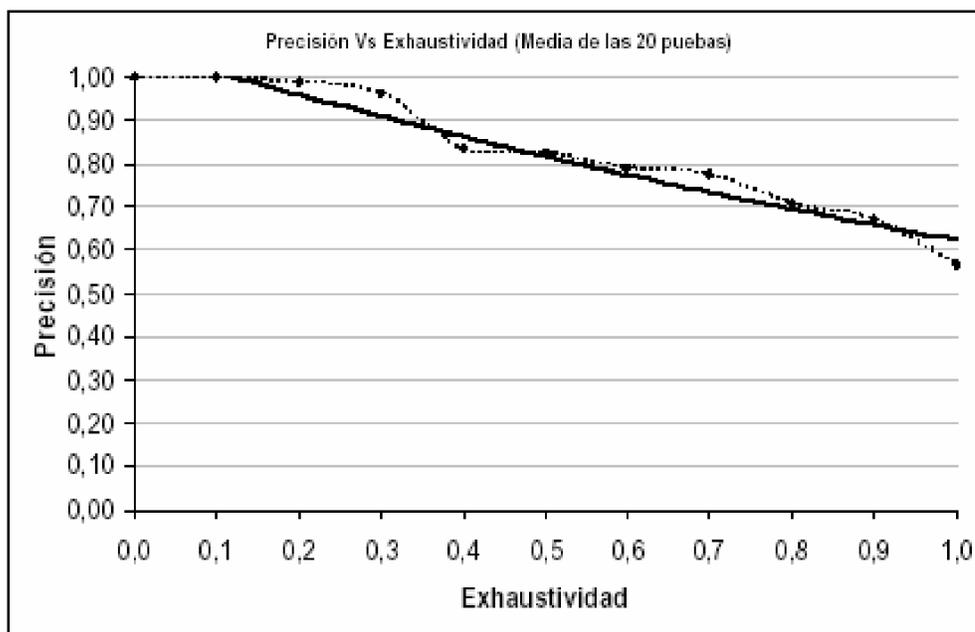
## Efectividad del sistema

Hechas las veinte pruebas se calcula la media para cada uno de los once valores de exhaustividad ver tabla 2. En la gráfica (figura 5) se puede observar que el sistema no soluciona al 100% el problema de la relevancia ya que para ello se necesita un valor de precisión de uno para los once valores de exhaustividad, también se observa que el comportamiento del sistema sigue la gráfica de Rijsbergen [RIJS 1999] figura 3, la cual dice que la exhaustividad se relaciona inversamente con la precisión.

**Tabla 2. Parejas Exhaustividad – Precisión para el sistema.**

Exhaustividad	Precisión
0	1,000
0,1	1,000
0,2	0,990
0,3	0,963
0,4	0,835
0,5	0,826
0,6	0,789
0,7	0,777
0,8	0,708
0,9	0,669
1,0	0,568

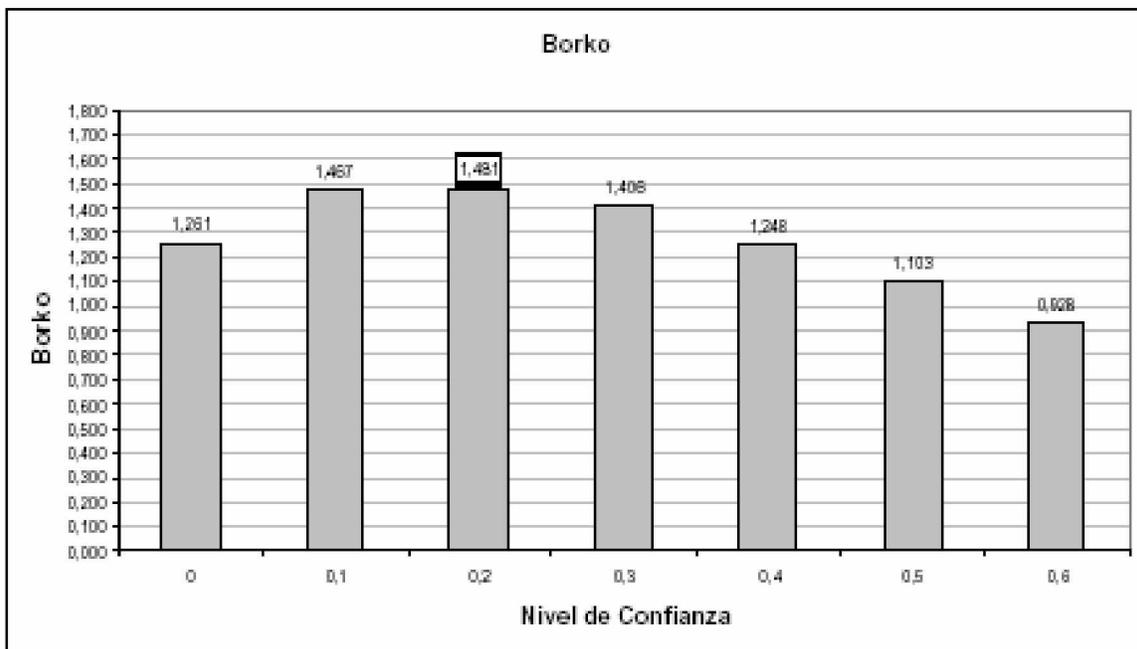
**Figura 5. Gráfica E-P para el sistema.**



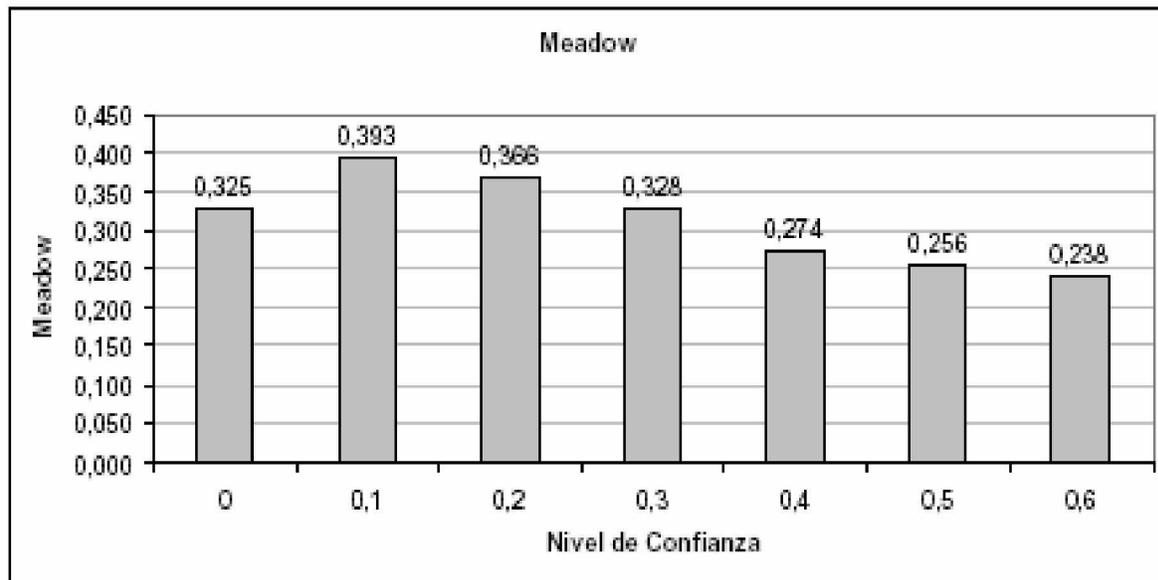
## EVALUACIÓN DE MEDIDAS ALTERNATIVAS

Con los datos de las veinte pruebas hechas se calculará el valor de las medidas alternativas vistas con el fin de obtener un nivel de confianza adecuado para mejorar la efectividad del sistema ya que las pruebas hechas anteriormente fueron hechas con un nivel de confianza de cero, es decir se presentaban todos los documentos recuperados por los agentes de filtrado y por lo tanto se recuperaban muchos documentos que por su bajo nivel de confianza eran no relevantes.

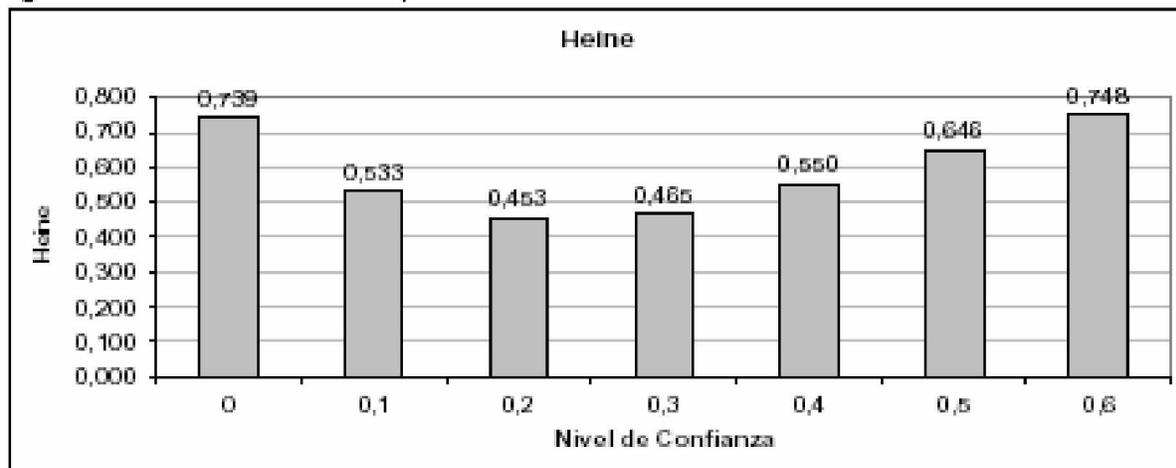
### Medida de Borko



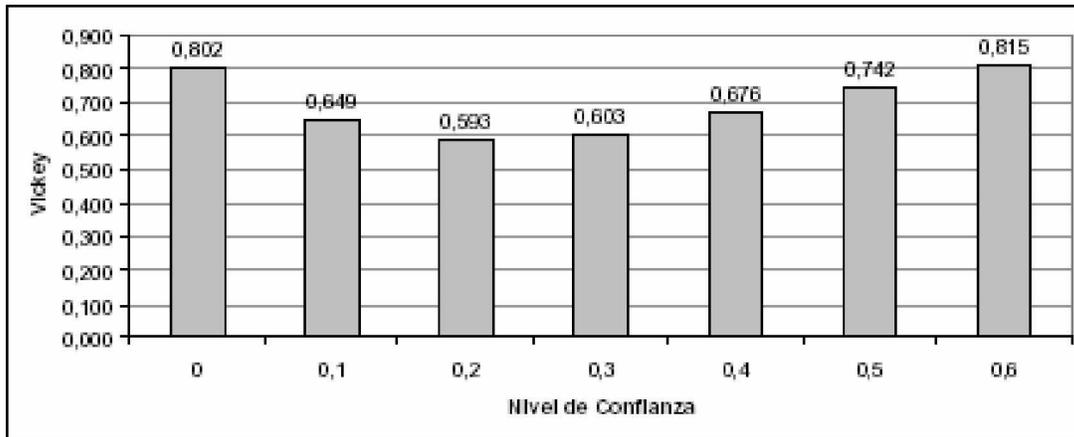
## Medida de Meadow



## Medida de Heine



## Medida de Vicky



La mayoría de las medidas alternativas sugieren que el sistema es más efectivo en un nivel de confianza mínimo de 0.2, éste valor determina entonces que documentos con un nivel de confianza mayor o igual que 0.2 pueden ser presentados al usuario de tal manera que se obtiene una mayor efectividad del sistema. La medida anterior se debe esencialmente a que la mayoría de documentos por debajo de 0.2 no son relevantes para el usuario lo que se evita una sobrecarga de información al mismo.

Se observa además de las anteriores gráficas de medidas alternativas un mal funcionamiento del sistema de recuperación cuando el nivel de confianza es alto (especialmente los niveles 0.5 y 0.6), lo anterior indica que al evolucionar los agentes con calificaciones positivas hechas por el usuario se omiten o excluyen documentos de la colección que pueden ser relevantes lo que produce una exhaustividad del sistema demasiado baja. Estos niveles mayores presentan una precisión alta aunque algunas veces dejen de recuperar documentos relevantes.

También se observa que al dejar un nivel de confianza de cero el sistema no funciona bien puesto que la precisión en este caso estaría disminuyendo, ya que se recuperarían documentos irrelevantes para el usuario. El nivel de confianza en cero también sugiere que el sistema sería bastante exhaustivo pero

poco preciso. Si bien la evolución ayuda a que el sistema se adapte al usuario, presenta el inconveniente de tener una exhaustividad muy limitada al punto en que no se pueda recuperar ningún documento.

## **CONCLUSIONES Y RECOMENDACIONES**

### **Conclusiones**

Durante el desarrollo del trabajo se evidenció que hay diversas teorías sobre los agentes inteligentes lo que hace que muchos autores difieran en la concepción de lo que verdaderamente es un agente, por esto existen diferentes herramientas para el diseño y desarrollo de sistemas multiagente las cuales no se relacionan de una manera adecuada entre sí. Se piensa entonces que falta una organización en el campo de los agentes inteligentes que unifique los aportes que han hecho los grupos de investigación e investigadores en un estándar, de tal manera que se presente una metodología y una herramienta de desarrollo para la construcción de SMA.

Según las pruebas y resultados obtenidos el prototipo propuesto supera en un gran margen al modelo de recuperación vectorial.

Aunque el prototipo desarrollado usa el modelo vectorial para la recuperación, representación e indexación de documentos, se obtienen mejores resultados debido al correcto funcionamiento de los agentes de filtrado los cuales representan de una manera adecuada los intereses del usuario presentando así a los documentos más relevantes en las primeras posiciones, de esta manera se puede asegurar que la representación de cada palabra clave y su peso en un documento es adecuada para que los sistemas de recuperación de información sean más efectivos.

El resultado de la comparación con otro sistema multiagente para la recuperación de información fue algo equilibrado, sistema multiagente WebFerret presenta mejores resultados ante valores de exhaustividad más altos, lo anterior nos indica que WebFerrert es mejor que el sistema desarrollado en este trabajo, la superioridad se debe talvez a que el sistema multiagente consultado recupera información de muchos motores de búsqueda lo que le permite ser más exhaustivo manteniendo una precisión alta.

Se logro encontrar un valor para el nivel de confianza de 0.2 que lo confirman las medidas alternativas de Borko, Heine y Vickey que produce mejores resultados ante el problema de la efectividad. Este valor al parece es un nivel adecuado en donde se recuperan la mayoría de documentos relevantes sin recuperar demasiados documentos irrelevantes lo que produce que se maximice la precisión y exhaustividad.

El ecosistema de agentes ayuda a una correcta adaptabilidad del sistema cuando el usuario va cambiando de intereses y califica negativamente a los agentes de filtrado de modo que estos van desapareciendo, sin embargo cuando se califica positivamente a los agentes se produce un incremento en el nivel de confianza llevando al sistema a tener una exhaustividad baja y por lo tanto produciendo un mal funcionamiento del mismo. El principal error de aplicarlo como lo hizo este trabajo es que un agente puede terminar con un genotipo con muchas palabras las cuales no se relacionarían entre sí y un fitness alto, lo que hace que un agente no encuentre documentos pertinentes al aumentar su nivel de confianza, es decir excluye documentos que pueden ser relevantes disminuyendo la exhaustividad del sistema. No obstante esta técnica de hacer evolucionar agentes inteligentes se puede usar para otras áreas de investigación ó de una manera diferente en sistemas de recuperación de información.

Los agentes diseñados y desarrollados permiten una recuperación de información acertada ante las palabras de interés y también un funcionamiento adecuado del sistema.

## **Recomendaciones**

La herramienta desarrollada corresponde a la clase de recuperadores de información, cuyo fin es ayudar a la recuperación de información relevante para el usuario. La información por lo general se encuentra en diferentes ubicaciones geográficas, está es una limitación del sistema desarrollado ya que solo se hace la recuperación en un solo lugar (localmente). La respuesta de los recuperadores de información por lo general es rápida, este prototipo es un poco más lento en la presentación de los resultados debido a que la indexación de los documentos se hace cada vez que se ejecuta el programa, los recuperadores utilizan una base de datos ya indexada de los documentos. Es necesario desarrollar dicha base de datos indexada con el propósito de presentar resultados más rápidos y también para que el prototipo implementado se pueda mejorar y migrarlo a una arquitectura cliente/servidor o aplicación Web ya que por el momento recupera los datos localmente.

El trabajo desarrollado puede servir como base para la implantación de una asignatura de profundización en Inteligencia Artificial Distribuida ya que presenta una descripción clara de los temas que se deben conocer para desarrollar sistemas multiagente y además presenta como anexos la metodología utilizada para el análisis y diseño de sistemas multiagente y la descripción de la herramienta con la que se pueden implementar.

Los agentes de filtrado junto con la representación de los intereses del usuario fueron claves para tener mejor efectividad que el modelo vectorial, se sugiere seguir investigando en estos dos aspectos junto con diferentes modelos para representar, indexar y recuperar información con el fin de obtener mejores resultados ya que al parecer el modelo vectorial de recuperación de información usado en el trabajo no es muy bueno en cuanto a efectividad en los sistemas de recuperación.

## Bibliografía

- [1] [ALON 1999] ALONSO Eduardo. Inteligencia Artificial Distribuida: Como entenderla y usarla, University of York. [on-line] <http://perso.wanadoo.es/jjreina/divulgacion/100tifica/index.htm>
- [2] [BENG 2001] BENG Ang Chwee y SHIMON Y. Nof. Desing Issies for Information Assurance with Agents: Coordination Protocols and Role Combination in Agents. Purdue University 2001.
- [3] [BOND 1988] BOND A.H. y GASSER L. Reading in Distributed Artificial Intelligence. San Mateo, Morgan Kaufmann 1988.
- [4] [CORC 1999] CORCHADO Juan, Agencia: Una puerta hacia la convergencia de la Inteligencia Artificial. Universidad de Salamanca 1999
- [5] [GREE 1997] GREEN Shaw, SOMERS Fergal y otros, Software Agent: A Review, Broadcom Ireland & Trinity College Dublin
- [6] [IGLE 1998] IGLESIAS Fernández Carlos. Definición de una Metodología para el Desarrollo de Sistemas MultiAgentes. Madrid, Universidad Politécnica de Madrid, 1998.
- [7] [INGE] INGENIAS. [on-line] [http://grasia.fdi.ucm.es/ingenias/index\\_eng.htm](http://grasia.fdi.ucm.es/ingenias/index_eng.htm)
- [8] [JENN 1995] JENNINGS Nicholas R., y WOOLDRIDGE Michael. Intelligent Agents: Theory and Practice, Londres, The Knowledge Engineering Review. 1995
- [9] [JENN 1996] JENNINGS N.R. Coordination Techniques for Distributed Artificial Intelligence, Londres, University of London, 1996.
- [10] [LANC 1993] Lancaster, F. W. and Warner, A.J. Information Retrieval Today. Arlington, Virginia , 1993.
- [11] [LUBO 2002] LUBOMIROVA Marinova Zlatina. Planning in Multiagent Systems. Sofia, Universidad de Sofia, 2002.
- [12] [MART 2002] MATINEZ Méndez Francisco Javier. Propuesta y desarrollo de un modelo para la evaluación de la recuperación de información en Internet. Universidad de Murcia 2002.

- [13] [NWAN 1996] NWANA Hyacinth S. Software Agents: Un Overview, The Knowledge Engineering Review
- [14] [OCHO 2004] OCHOA Avendaño Nelson Daniel. Implementación de un sistema prototipo para la recuperación de información en documentos digitales utilizando un ecosistema de agentes inteligentes. Bogotá, Universidad Nacional de Colombia 2004.
- [15] [RIJS, 1999] Rijsbergen, C.J. Information Retrieval. [on-line]. Glasgow, University, 1999. <http://www.dcs.gla.ac.uk/~iain/keith/>
- [16] [SAL, 1983] Salton , G. and Mc Gill, M.J. *Introduction to Modern Information Retrieval*. New York: Mc Graw-Hill Computer Series, 1983.
- [17] [WOOL 1994] WOOLDRIDGE Michael, Coherent Social Action, ECAI 94 11th European Conference on Artificial Intelligence, John Wiley & Sons Ltd 1994.
- [18] [WOOL 1995] WOOLDRIDGE Michael y JENNINGS Nicholas. Agent Theories, Architectures and Languages: A Survey. ATAL 1995.