

# APLICACIÓN DE METODOLOGÍA GAIA<sup>1</sup> A UN SISTEMA MULTIAGENTE DE FILTRADO COGNITIVO Y SOCIAL

María Helena Mejía  
Oscar Franco

**Palabras Clave:**

Agente, multiagentes, autonomía, sociabilidad, reactividad, pro-actividad

## PLANTEAMIENTO DEL PROBLEMA

La introducción de la tecnología de agentes en la industria requiere de nuevas metodologías que asistan en todas las fases del ciclo de vida en el desarrollo de un sistema de agentes. En la actualidad esta tecnología está recibiendo una gran atención y esfuerzo. Sin embargo, a pesar del rápido desarrollo de teorías, arquitecturas y lenguajes de agentes se ha realizado muy poco trabajo en la especificación y aplicación de técnicas para crear aplicaciones empleando dicha tecnología. Las metodologías existentes son estructuradas pero no tienen un enfoque formal para el análisis y diseño de los sistemas de agentes, y las metodologías que predominan son las inspiradas en los modelos orientados a objetos; es

(1) El nombre proviene de la hipótesis Gaia propuesta por el científico británico James Lovelock en honor a la antigua Diosa de la tierra. La hipótesis o también conocida como la teoría GAIA dice en términos generales que todos los organismos de la biosfera de la tierra actúan juntos para regular el medioambiente de esta.



quizás la metodología GAIA propuesta por Wooldridge, Jennings y Kinny (2000).

Por otro lado debido al creciente y acelerado uso de la Internet por las características dominantes en la infoesfera: localizar fuentes de información, accederla, filtrarla e integrarla se han convertido en tareas críticas. La teoría de agentes es adecuada y promisoría para ayudar a resolver este problema, como lo han mostrado los desarrollos basados en agentes para la Internet. Algunos de ellos son considerados asistentes personales y una de sus funciones es recomendar información de manera proactiva a un usuario, basándose en técnicas de filtrado colaborativo y filtrado cognitivo. En realidad muchos de los sistemas de agentes de Web tienen algún grado de conocimiento del usuario, esta inclusión de información se está convirtiendo rápidamente en un área clave tal como lo propone la World Wide Web Consortium (1997, Junio 02) en su reporte técnico *Proposal for an open profiling*.

## OBJETIVO GENERAL

Implementar un prototipo de un sistema orientado a agentes para el filtrado cognitivo y colaborativo siguiendo la emergente metodología GAIA. Este sistema debe adicionar inteligentemente a los *bookmarks* de un usuario las páginas recuperadas de la red, en los temas y subtemas que defina en sus preferencias, además debe recomendar estas páginas a otros usuarios del sistema que tengan intereses similares.



## CONCEPTOS BÁSICOS

### *Agentes y multiagentes*

Según Rusell y Norvig (1995) un agente es algo que puede ser visto percibiendo el entorno y actuando sobre él. Esta definición depende fuertemente de lo que se considere como "entorno" y también de lo que se considere como "percibir" y "actuar". Un agente es una entidad software con un propósito específico. La persistencia distingue a los agentes de las ya clásicas subrutinas, ya que los agentes tienen sus propias ideas sobre cómo realizar tareas y las subrutinas no. El propósito específico los distingue de las aplicaciones generales, aunque ciertamente hay autores que no consideran el tener un propósito específico como una de las características fundamentales de los agentes. Con base en el trabajo de Iglesias (1997), definimos lo siguiente:

### **Características**

**Autonomía:** Los agentes pueden operar sin la intervención de humanos o de otros agentes. Los agentes encapsulan algún estado y toman decisiones en lo que deben hacer basados en dicho estado. La autonomía es una de las características más importante del concepto de agente.

**Sociabilidad:** Los agentes son capaces de interactuar con otros agentes (humanos o no) a través de un lenguaje de comunicación entre ellos y típicamente tienen habilidades para emprender actividades sociales como la negociación o la capacidad de solucionar problemas en forma cooperativa con el fin de alcanzar las metas.



**Reactividad:** Los agentes están situados en un entorno y son capaces de percibir estímulos de allí y reaccionar a los estímulos en corto tiempo, es decir, tienen la habilidad de responder mientras el cambio ocurre.

**Pro-actividad:** Es una de las consecuencias de la autonomía de un agente. Este es capaz de elegir, en cada momento, cuales son las acciones a llevar a cabo para alcanzar sus objetivos. Es decir, no solo actúa en función de los estímulos que recibe desde el exterior, sino que puede realizar acciones como resultado de sus propias decisiones.

La definición de lo que es un agente, aún sigue siendo ampliamente debatida y algunos investigadores en especial los que trabajan con Inteligencia artificial consideran que las anteriores no son suficientes para caracterizar a los agentes y proponen otras adicionales tales como las que plantean (Shoham, 1993; Bates, 1994): veracidad, benevolencia, inteligencia, racionalidad, coherencia, adaptabilidad y movilidad.

### Sistemas multiagente

El paradigma de los Sistemas Multiagente (MAS) se fundamenta en la idea de superar las limitaciones inherentes a cualquier sistema inteligente, natural o artificial, formando agrupaciones de sistemas simples de tal forma que puedan compartir conocimientos y capacidades de la misma manera que la gente supera sus limitaciones individuales agrupándose en sociedades. Cada uno de estos sistemas simples que agrupamos constituiría un agente y la comunidad considerada globalmente es un sistema multiagente.



La nueva forma de plantear los problemas que ofrecen los MAS es clara y simple, ya que es una solución que la naturaleza ha aplicado desde siempre.

## Metodología GAIA

La metodología GAIA (1) como lo propone Wooldrigdge, Jennings y Kinny (2000) permite ir sistemáticamente desde los requerimientos hasta un diseño lo suficientemente detallado que puede ser implementado directamente. El análisis y el diseño se pueden ver como el proceso de desarrollo incremental que conlleva a modelos más detallados del sistema que se va a construir. Los principales conceptos de gala se pueden dividir en dos categorías: abstractas y concretas. Las entidades abstractas son las que durante el análisis, conceptualizan el sistema pero no necesariamente tienen una realización directa en el sistema. En contraste las entidades concretas son usadas en el proceso de diseño y típicamente están más relacionadas con el sistema en tiempo de ejecución.

Un rol está definido por cuatro atributos: responsabilidades, permisos, actividades y protocolos.

Gráfico 2. Conceptos abstractos y concretos en GAIA

Conceptos Abstractos	Conceptos Concretos
Roles	Tipos de Agentes
Permisos	Servicios
Responsabilidades	Conexión/Comunicación
Protocolos	
Actividades	
Propiedades de Vivacidad	
Propiedades de Seguridad	

El modelo de análisis se divide en tres fases: Requerimientos, Modelo de roles y Modelo de interacciones y el proceso de diseño involucra tres modelos: el modelo de agentes, el modelo de servicios y el modelo de familiaridad.

### Recuperación de información

El problema de recuperación de información según se puede definir como: dada una necesidad de información de un usuario y un conjunto de documentos, ordenar los documentos de más a menos relevantes para esa necesidad y presentar un subconjunto de los más relevantes. Las etapas para abordar el problema son:

- Elegir un modelo que permita calcular la relevancia de un documento frente a una consulta.
- Diseñar algoritmos y estructuras de datos que lo implementen eficientemente como los índices.

El recuperado de información presentado por Belkin y Croft (1992), requiere modelos y algoritmos especializados. Los

Gráfico 3. Relaciones existentes entre los modelos de GAIA.



modelos se enfocan principalmente en el proceso de comparación. Los principales modelos o los modelos clásicos son: el booleano, el del vector y el probabilístico.

### Filtrado de información

El filtrado de información involucra dos aspectos básicos: grandes conjuntos de datos no estructurados o datos textuales, y necesidades de información que son relativamente estables a través del tiempo.

### DESCRIPCION DEL SISTEMA MULTIAGENTE

El funcionamiento general del sistema planteado por Mejía y Franco (2004) es el siguiente:

#### Gráfico 4. Modelo de Interacción.

Un ejemplo de modelo de interacción.- Protocolo autorizar Usuario Nuevo

<b>Nombre protocolo:</b> SolicitarAutenticaciónUsuario
<b>Rol Emisor:</b> Usuario Autenticador
<b>Rol Receptor:</b> Coordinador
<b>Entradas:</b> (Identificación, Contraseña)
<b>Salidas:</b> Trae
<b>Procesamiento:</b> El rol autenticador toma los datos del usuario : Identificación y Contraseña y los envía al usuario coordinador para su validación.

<b>Nombre protocolo:</b> EnviarAutenticaciónUsuario
<b>Rol Emisor:</b> Coordinador
<b>Rol Receptor:</b> Usuario Autenticador
<b>Entradas:</b> (resp) y (resp = True si existe usuario o resp = False si no existe usuario)
<b>Salidas:</b> trae
<b>Procesamiento:</b> El rol Coordinador verifica en la base de datos la existencia de un usuario con los datos enviados. Le envía la respuesta obtenida, al rol receptor.

<b>Procesamiento:</b> El rol Coordinador verifica en la base de datos la existencia de un usuario con los datos enviados. Le envía la respuesta obtenida, al rol receptor.
--



- ♦ La primera vez que un usuario entra al sistema debe registrarse, es decir, escoger un nombre de usuario y una clave de acceso (identificación, contraseña). Además debe proporcionar información acerca de sus intereses, los cuales serán obtenidos mediante un cuestionario.
- ♦ La información que proporciona el usuario, inicialmente, es el primer paso en el proceso de adaptación automática del sistema a las preferencias del usuario y al modelo que permitirá elaborar las entradas a los temas principales que conforman sus documentos preferidos o "bookmark".
- ♦ Los accesos siguientes al sistema por parte de un usuario, únicamente requieren la introducción del nombre de usuario y clave de acceso.
- ♦ El sistema debe proporcionarle al usuario una opción en la

Gráfico5





Tabla N° 6

SERVICIO DEL AGENTE COORDINADOR: AutenticarUsuario
Entradas: IdUsuario , Contraseña
Salidas: esVálidaAutenticación
Precondiciones: esValidoFormulario =True
Poscondiciones: si ( (idUsuario, Contraseña) $\in$ Tabla Datos Personales) entonces esVálidaAutenticación =True si ( (idUsuario, Contraseña) $\notin$ Tabla Datos Personales) entonces esVálidaAutenticación =False

cual pueda observar las nuevas sugerencias a través de páginas URL` s enviadas por otros usuarios ya registrados en el sistema y quienes tienen intereses similares.

- ♦ El usuario tendrá la opción de aceptar o rechazar las sugerencias o recomendaciones realizadas por otro usuario.
- ♦ El sistema debe permitirle al usuario consultor ver todas las sugerencias que le hayan realizado todos los usuarios, y cada

Tabla 7. Modelo de familiaridad



sugerencia debe contener la fecha en que fue enviada, fecha de aceptación o rechazo por parte del usuario consultor y el respectivo estado: rechazada o aceptada.

- ♦ El usuario a través del sistema debe poder navegar en Internet y cuando el usuario se encuentre interesado en grabar un URL, el sistema automáticamente le propondrá una clasificación según los temas que previamente el usuario había registrado. El usuario tendrá la opción de aceptar la categoría propuesta o rechazarla. En caso de rechazo, el usuario podrá clasificar el documento e ingresarlo a la categoría respectiva, de forma manual.

Gráfico 8. Diagrama modular

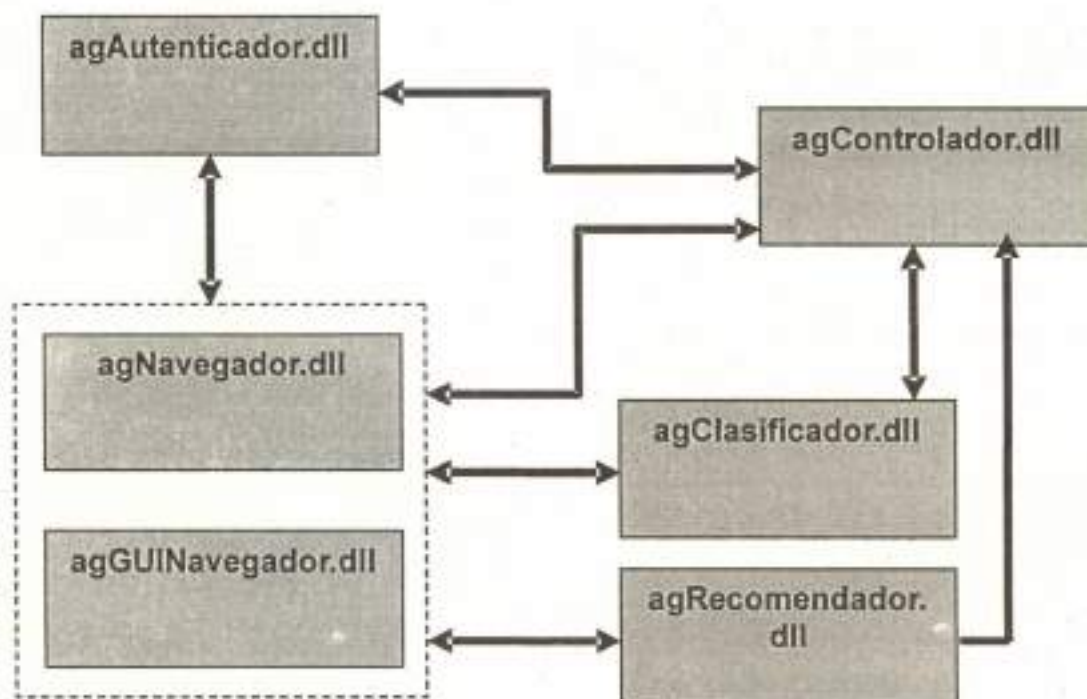


Gráfico9

<b>SERVICIO DEL AGENTE USUARIO: SolicitarGrabarDatosRegistroUsuario</b>
<b>Entradas:</b> IdUsuario, contraseña, nombreUsuario, listadoPreferencias, esValidoRegistroUsuario
<b>Salidas:</b> IdUsuario, contraseña, nombreUsuario, listadoPreferencias, esValidoRegistroUsuario
<b>Precondiciones:</b> esValidoRegistroUsuario=true
<b>Poscondiciones:</b> true

- ♦ Una vez se ha clasificado el documento, el sistema se encargará automáticamente de enviarlo a los demás usuarios del sistema que tengan gustos similares.

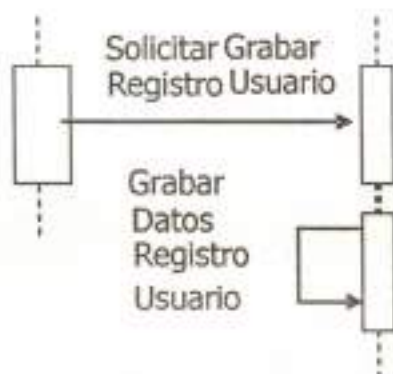
## APLICACIÓN DE LA METODOLOGIA GAIA EN EL PROTOTIPO

### ANALISIS

1. Identificar los roles claves del sistema. Un rol es una abstracción en el modelo de la organización, en el cual todo el sistema se puede ver como dividido en roles.

Salida: Un prototipo de modelo de roles. Una lista de roles claves que ocurren en el sistema, cada uno con una descripción informal.

Gráfico10



2. Para cada rol, identificar y documentar cada protocolo asociado.

Salida: Un modelo de interacción, los cuales capturan los patrones de la interacción entre roles.

3. Elaborar el modelo de roles.

Salida: Un modelo de roles completamente elaborado, el cual documenta los roles que ocurren en el sistema, sus permisos y responsabilidades, con los respectivos protocolos y actividades en los cuales ellos participan.

4. Iterar los pasos 1-3 hasta obtener el refinamiento deseado.

### **Roles del sistema**

En la primera iteración del análisis, estos son los roles identificados:

**Rol usuario autenticador:** Es el encargado de identificar a los usuarios del sistema mediante la validación de los datos de ingreso (Identificación, clave) y el ingreso de los datos que forman el perfil inicial para los nuevos usuarios.

**Rol navegador:** Este rol es el que suministra la información mas importante para actualizar los favoritos o bookmarks de un usuario, debido a que es el encargado de los procesos de interfase del usuario con la Web, mediante su interacción con el navegador.

**Rol autorizador:** La función de este rol es identificar las acciones de los usuarios con respecto a su aceptación o rechazo de: categorías para documentos realizadas por el



sistema o por el grupo de usuarios vecinos con intereses similares.

## DISEÑO

### Modelo de agentes

Para el modelo de agentes del sistema se encapsularon todos los roles en cuatro agentes: Usuario, Coordinador, Clasificador y Recomendador. Con el fin de optimizar el diseño de acuerdo a las principales características del sistema y mejorar la eficiencia en la implementación, aunque todas las características identificadas en los roles deben ser implementadas. Pero al agruparlas se mejora la comprensión del sistema y además se optimiza el uso de los diferentes recursos. Los roles Autenticador Navegador y Autorizador tienen un alto grado de interdependencia al igual que los roles Filtrador cognitivo e Indexador, como también el rol Recomendador y el Filtrador social por lo tanto se agrupan como se especifica en la gráfica 5.

### Modelo de servicios

Uno de los 73 servicios que modelan el sistema, se ilustra en la siguiente tabla N° 6

La metodología GAIA, permite diseñar todo el sistema Multiagente de manera completa, sin embargo el grado de abstracción es alto, lo que dificulta la implementación directamente. Para ello se hace necesario aplicar otros artefactos de modelado que faciliten la implementación como lo propone Arenas, García y Pérez (2004). *AUML* es una herramienta que permite modelar sistemas basados en agentes usando un conjunto de extensiones del *UML*. El



trabajo de Odell, Parunak y Bauer (2000) representa *AUML* de tres niveles para protocolos de interacción de agentes. El *AIP* (*Agent Interaction protocol*) describe un patrón de comunicación como una secuencia válida de mensajes entre agentes y las restricciones sobre los contenidos de estos mensajes. Estos tres niveles de *AIP* representan básicamente: el protocolo como un todo, transacciones o interacciones entre agentes y la actividad interna de los agentes.

*AUML* se utilizó de una manera diferente rescatando los elementos fundamentales del sistema identificados con GAIA y particularmente a partir del diseño obtenido de tal metodología.

## IMPLEMENTACION DEL SISTEMA

### *Requerimientos técnicos*

#### **Plataforma de ejecución:**

- Sistema operativo Windows XP Windows 2000 en adelante.
- Plataforma *sdk* redistribuible para ejecutar aplicaciones en .Net

#### **Plataforma de agentes:** JADE

#### **Plataforma de desarrollo:** C# para .Net

### *Descripción general de los módulos*

El sistema esta conformado por 4 módulos, cada uno de los cuales corresponde con un agente del sistema, de acuerdo al modelo de agentes obtenido aplicando la metodología GAIA. Los agentes del sistema son: Agente Usuario, Agente Coordinador, Agente Clasificador y Agente Recomendador. A cada uno de los cuales le corresponde una DLL desarrollada en



c#, sin embargo para el agente Usuario se crearon dos módulos el Autenticador y el Navegador ya que por su funcionalidad no es necesario mantener todos los servicios del agente Usuario Activos, lo cual permite que el sistema se ejecute con mayor eficiencia al disminuir los hilos de ejecución que deben mantenerse activos.

El punto de partida para la implementación de los módulos fue la definición de los servicios del modelo GAIA en la que se describen las pre y post de los módulos que fueron de gran ayuda para los algoritmos básicos como: captura de datos, validaciones, etc. Los diagramas de secuencia y de actividad de las capas de AUML, se utilizaron para la implementación de la comunicación entre agentes y para los algoritmos complejos. Cada servicio y/o cada acción especificada en el diseño de GAIA esta implementada en un comportamiento del sistema.

### Ejemplo

En el modelo de servicios del Agente usuario se tiene el siguiente servicio:

En el modelo en AUML se registra este servicio con el siguiente protocolo

La implementación del servicio Solicitar Grabar registro Usuario esta registrada en el agente JADE Usuario Autenticador como un comportamiento del agente.

```
public SgduBehaviour(Agent a) : base(a)
{
}
```

que hace parte de un comportamiento compuesto FMS definido en JADE.



En el metodo action del comportamiento se envía el mensaje al agente Coordinador utilizando *ACL (Agent Communication language)*.

```

:::::
AID rcpt = new AID();
rcpt.setName("agControlador@"+dirIp+":1099/JADE");
A C L M e s s a g e      m s g      =      n e w
ACLMessage(ACLMessage.REQUEST);
msg.setReplyWith("GRABAR USUARIO");
msg.setSender(a.getAID());
msg.addReceiver(rcpt);
msg.setContentObject(dP);

```

## CONCLUSIONES

La construcción del sistema siguió todos los pasos de la metodología GAIA. Se dio inicio en la fase de análisis con la definición de los roles que harían parte de la organización; luego se identificaron los protocolos, es decir, las interacciones entre los roles para obtener el respectivo modelo de interacción basados inicialmente en "super protocolos" además como complemento se elaboró un diagrama de relaciones entre roles y por último se construyó el esquema de roles ofreciendo una vista mas formal de estos. Toda la fase de análisis se realizó en dos ciclos obteniendo obviamente mejores resultados en el segundo. Para la fase de diseño se elaboró el modelo de agentes que representa a aquellos que van a ser utilizados en el sistema con las respectivas instancias que serían utilizadas en tiempo de ejecución; luego se identificaron los servicios asociados con cada rol de agente y se especificaron las propiedades principales de estos con el fin de construir el modelo de servicios (funciones de los agentes). Por último se





generó el modelo de familiaridad/comunicación el cual define gráficamente el vínculo comunicativo entre un agente y otro.

Como los resultados GAIA en la etapa de diseño son todavía especificaciones a muy alto nivel y no permite llegar directamente a una fase de implementación, siguiendo el trabajo de Arenas, García y Pérez (2004), se propuso el uso de *AUML*. Con *AUML* se pueden detallar aspectos de un sistema basado en agentes usando el *AIP* y lograr facilidad para la implementación. *AIP* (*Agent Interaction Protocol*) permite generar una representación a tres niveles usando extensiones de *UML*, los niveles son: nivel de protocolos, nivel de interacción y nivel de proceso interno de los agentes. Para el prototipo planteado se desarrollaron los tres niveles propuestos del *AIP*, no se partió de cero pues se retoman los elementos fundamentales del sistema identificados con GAIA y particularmente a partir del diseño obtenido de tal metodología.

GAIA busca especificar cómo una sociedad de agentes colabora para alcanzar los objetivos del sistema, y qué se requiere de cada uno para lograrlo. El principal inconveniente de GAIA es que se queda a un nivel de abstracción demasiado alto. Según los autores, con ello se consigue desacoplarse de las distintas soluciones de implementación de agentes. Por lo tanto es una buena combinación la de GAIA y *AUML*, pues con el último se consigue el nivel de abstracción bajo para pasar a la implementación y se conserva el desacople.

La metodología GAIA en la parte de análisis, presenta un enfoque natural y bastante eficiente para obtener el modelo de roles y el de protocolos aunque es deficiente el modelo de interacciones ya que no plantea un diagrama que permita tener



una visión holística y detallada de todas las interacciones entre los roles. En la parte de diseño, una debilidad de GAIA derivada de la alta abstracción es que no proporciona elementos para especificar un servicio interno o tarea de un agente. Por último, el modelo de familiaridad/comunicación de GAIA es bastante simple y no otorga un mayor valor agregado al diseño.

Para la implementación se utilizó la plataforma *middle-ware* de desarrollo de sistemas multiagente - JADE ya que: es bastante extendida y soporta todos los servicios básicos de infraestructura especificados en FIPA (comunicaciones, movilidad, gestión de agentes y localización de agentes), y a los que añade algunas utilidades gráficas para facilitar la administración de las plataformas y la depuración de los mensajes intercambiados por agentes en tiempo de ejecución. Para la implementación de agentes con esta plataforma se utilizan los estándares de comunicación y de gestión de agentes. El resto, como la especificación de los servicios del agente, las relaciones entre las tareas del sistema, se implementaron en c# de la plataforma *dotnet* de Microsoft. Generalmente los desarrollos en JADE van acompañados de java, pero la selección de la plataforma de .net obedece mas a la novedad y a preferencias personales que a criterios realmente técnicos.

El prototipo del sistema implementó todos los servicios, funciones y protocolos definidos en las etapas de análisis y diseño, cada uno de ellos se define como un comportamiento en los agentes JADE. El cual se basa en un sistema multihilos que permite el funcionamiento concurrente de todos los agentes.



## REFERENCIAS

Arenas, A.E. y García-ojeda, J.C. y Pérez-Alcázar, J.J. (2004, Marzo). *On combining organisational modelling and graphical languages for the development of multiagent systems*. IOS Press: Integrated Computer-Aided Engineering

Bates, J. (1994). *The role of emotion in believable agents*. Communications of the ACM

Belkin N.J y Croft W.B. (1992). *Information filtering and information retrieval: Two sides of the same coin?*. Communications of the ACM

Iglesias C.A. (1997). *Fundamentos de los sistemas inteligentes*. Informe técnico UPM/DIT/GSI. Departamento Ingeniería de sistemas telemáticos, E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid.

Mejía, M.H y Franco, O. *Prototipo de un Sistema Multiagente de filtrado cognitivo y colaborativo bajo la metodología Gaia*. 2004. Tesis de maestría. Universidad Autónoma de Bucaramanga en convenio con el Instituto Tecnológico de Monterrey.

Rusell, S. y Norvig P. (1995) *Artificial intelligence: A modern approach*. 1ra. Edición.

Shoham, Y. ) (1993, Marzo); *Agent-oriented programming*. Artificial intelligence.



Wooldridge, M y Jennings, R. y Kinny, D. (2000), *The GAIA methodology for agent-oriented analysis and design*. International journal of autonomous agents and multi-agent systems, 3(3)

World Wide Web Consortium (1997, Junio 02) *Proposal for an open profiling*. Extraído el 14 de Mayo de 2002 desde <http://www.w3.org/TR/NOTE-OPS-FrameWork>

