

**UNIVERSIDAD MAYOR DE SAN ANDRES  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMATICA**



**TESIS DE GRADO**

**COMPARACION DE METODOLOGIAS MAS-commonKads E INGENIAS  
EN EL DESARROLLO DE SISTEMAS MULTI-AGENTE.  
CASO DE ESTUDIO: SISTEMA MULTI-AGENTE DE LICITACIONES**

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA  
MENCION INGENIERIA DE SISTEMA INFORMATICOS**

**POSTULANTE: Univ. Lucía Laura Tonconi**

**TUTOR: Lic. Eufren Llanque Quispe**

**REVISORA: Lic. Brígida Carvajal Blanco**

**LA PAZ – BOLIVIA**

**2009**

## **Dedicatoria**

*A Dios*

*Que permitió que este momento llegue  
Justo en el tiempo y espacio que tenía que suceder.  
Por que las cosas siempre pasan por algo  
Y de esos momentos estamos hechos*

*A mi familia*

*Por todo su apoyo, comprensión y desprendimiento  
Por su amor, por su fortaleza  
Por que ellos tenían que ser mi familia  
Por que esas dos personas tenían que ser mis padres  
Y ellos tenían que ser mis hermanos  
Para: Concepción, Santos, Ronald y Liliam*

*A mis mejores amigos*

*Juana, Marianela, Rolando, Gabriel, Luís,  
Ludbin, Gabriela, Elizabeth y Williams  
Gracias por su amistad y apoyo*

**E-mail: *lucy\_laurat@hotmail.com***

## *Agradecimientos*

Durante el largo recorrido que representa la vida académica, bajo distintas circunstancias se conoce a muchas personas que de alguna forma dejan huella, que se convierten en recuerdos de verdaderas amistades. Con las disculpas del caso y al pecar de omisión, a la conclusión del trabajo de tesis deseo agradecer en especial:

Al Lic. Eufren Llanque Quispe, por su asesoría, consejos y soporte lo cual que permitió culminar exitosamente el presente trabajo de tesis.

A la Lic. Brígida Carvajal Blanco, por su paciencia ante todo, por su orientación, asesoría, su profesionalismo, su amistad, su apoyo y sus palabras de aliento ante situaciones adversas, en cuanto al desarrollo y conclusión del trabajo de tesis. Muchas Gracias Licenciada.

Al Ing. Guillermo Callisaya, por que cuando uno quiere hacer algo lo primero que hace es empezar, gracias por tu tiempo, tu paciencia, ante todo tu amistad y ayudarme a empezar en Informática, no podía culminarla sin agradecerte.

Al Lic. Ludbin Guilberth Valencia, por su asesoramiento y consejos reflejados en el apoyo y voz de aliento para la definición y sustentación del trabajo de tesis.

A mis queridos padres Concepción Tonconi y Santos Laura por su apoyo incondicional para concluir el presente trabajo de Tesis. Y a mis hermanos Ronald Laura y Liliam Laura por su paciencia y por ayudarme a ser cada día mejor persona.

A mis compañeros de la Universidad en especial a mis mejores amigos Juana Sillo (Juanita), Rolando Clavel (Rolo), Gabriel Rivera (Gabo), Luís Silvestre, Gabriela Bustos (Gaby), Elizabeth Rondo (Ely), Williams Yujra (Will), y a una amiga que no esta mi lado, pero esta en corazón Marianela Luizaga (Mari) allá lejos en España, gracias amigos por estar ahí en diferentes circunstancias alegres y tristes, sin importar distancias ni tiempos. Compartir risas se lo hace con cualquier persona, pero compartir lágrimas solo con los mejores amigos. Gracias por su amistad y apoyo a lo largo de la vida académica.

## RESUMEN

En los últimos años el avance de la Inteligencia Artificial dirigida al desarrollo de Agentes Inteligentes y de Sistemas Multi-Agentes conllevó a la creación de más de veinte metodologías de Desarrollo de Agentes y Sistemas Multi-Agente, cada metodología se creó bajo diferentes enfoques, las reconocidas son el enfoque Orientada a Objetos, enfoque de Ingeniería de Conocimiento o de Teoría de Agentes. Actualmente no existe una solución que fuerce al uso en concreto de alguna de las metodologías, debido a que no existen técnicas de evaluación sujeto a normas vigentes para la construcción acertada de Sistemas Multi-Agente.

Existen trabajos relacionados a evaluar y comparar al menos diez metodologías, con la ayuda de muchos profesionales, realizando una encuesta en base a cuatro características, llegando a conclusiones en base a las experiencias que tuvo cada profesional al desarrollar un determinado Sistema Multi-Agente, pero esta comparación no fue hecha en base a un Sistema Multi-Agente en concreto.

En el presente trabajo de Tesis se estudia las características de dos de las metodologías más representativas como MAS-CommonKADS del enfoque de Ingeniería de Conocimiento cuanta con siete modelos, e INGENIAS creada bajo el enfoque Orientado a Objetos que esta constituida por cinco modelos, ambas aplicadas a un Caso de Uso en concreto un "Sistema Multi-Agente de Licitaciones". Se realiza la comparación de ambas metodologías bajo las características relacionadas al proceso, al modelo, a los pasos y a la característica alentadora de cada metodología y se da una respuesta en cuanto a si existen diferencias o similitudes significativas.

# INDICE

<b>CAPITULO I MARCO REFERENCIAL</b> .....	1
1.1. INTRODUCCION .....	1
1.2. ANTECEDENTES .....	2
1.3. PLANTEAMIENTO DEL PROBLEMA .....	5
1.3.1. PLANTEAMIENTO DEL PROBLEMA .....	5
1.4. OBJETIVOS DEL PROYECTO .....	5
1.4.1. OBJETIVO GENERAL .....	5
1.4.2. OBJETIVOS ESPECIFICOS .....	5
1.5. HIPOTESIS .....	6
1.6. JUSTIFICACION .....	6
1.6.1. JUSTIFICACION CIENTIFICA .....	6
1.6.2. JUSTIFICACION TECNICA .....	6
1.6.3. JUSTIFICACION SOCIAL .....	7
1.7. METODOLOGIA .....	7
1.8. ALCANCES Y APORTES .....	8
1.8.1. LIMITES Y ALCANCES .....	8
1.8.2 APORTES .....	8
<b>CAPITULO II. MARCO CONCEPTUAL</b> .....	9
2.1. AGENTES .....	9
2.1.1 AGENTES INTELIGENTES .....	10
2.2. SISTEMAS MULTI-AGENTE .....	11
2.2.1. FIPA (FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS) .....	12
2.2.2. ARQUITECTURA ABSTRACTA DE FIPA .....	12
2.2.3. FIPA ACL (AGENT COMMUNICATION LENGUAJE) .....	13
2.2.4. COMUNICACION ENTRE AGENTES .....	13
2.3. METODOLOGIAS PARA EL DESARROLLO DE SMA'S .....	15
2.4. METODOLOGIA MAS-COMMONKADS .....	16
2.4.1. PERSPECTIVA HISTORICA MAS-COMMONKADS .....	16
2.4.2 LIMITACIONES DE COMMONKADS PARA SISTEMAS MULTIAGENTE .....	16
2.4.3. MAS-COMMONKADS .....	17
2.4.4. CARACTERISTICAS DE MAS-COMMONKADS .....	17
2.4.5. FASES DE DESARROLLO DE MAS-COMMONKADS .....	18

2.4.6. MODELOS DE MAS-COMMONKADS .....	18
2.4.7. MODELO DE AGENTE.....	19
2.4.7.1. CICLO DE DESARROLLO DEL MODELO DE AGENTE.....	20
2.4.8. MODELO DE TAREAS .....	20
2.4.8.1. DESARROLLO Y GESTIÓN DEL MODELO DE TAREAS.....	21
2.4.9. MODELO DE EXPERIENCIA .....	21
2.4.9.1. ESTRUCTURA DEL MODELO DE EXPERIENCIA.....	22
2.4.9.2. TÉCNICAS DE ADQUISICIÓN DEL CONOCIMIENTO.....	22
2.4.10. MODELO DE COORDINACIÓN .....	22
2.4.10.1. ESTRUCTURA DEL MODELO .....	23
2.4.11. MODELO DE ORGANIZACIÓN.....	23
2.4.11.1. CICLO DE DESARROLLO DEL MODELO.....	24
2.4.11.2. ACTIVIDADES DE CONSTRUCCIÓN DEL MODELO DE ORGANIZACIÓN.....	24
2.4.12. MODELO DE DISEÑO.....	24
2.4.12.1. ESTRUCTURA DEL MODELO .....	25
2.4.13. MODELO DE COMUNICACIÓN.....	26
2.5. METODOLOGIA INGENIAS .....	27
2.5.1. CARACTERISTICAS DE INGENIAS .....	28
2.5.2. NOMENCLATURA .....	28
2.5.3. ENTIDADES BÁSICAS.....	29
2.5.4. NOTACIÓN .....	29
2.5.5. EL LENGUAJE DE ESPECIFICACIÓN .....	30
2.5.5.1. META-MODELOS .....	31
2.5.6. MODELO DE AGENTE.....	32
2.5.7. MODELO DE OBJETIVOS Y TAREAS .....	33
2.5.8. MODELO DE ORGANIZACIÓN.....	35
2.5.9. MODELO DE INTERACCIONES.....	36
2.5.10. MODELO DE ENTORNO.....	37
2.5.11. INGENIAS DEVELOPMENT KIT (IDK).....	38
2.6. JADE .....	40
2.6.1. ARQUITECTURA.....	41
2.6.2. PROGRAMACIÓN DE AGENTES EN JADE.....	41
2.6.3. RAZONES PARA EL USO DE JADE .....	42
<b>CAPITULO III. DESARROLLO DE SMA DE LICITACIONES.....</b>	<b>43</b>
3.1. DEFINICION CASO DE ESTUDIO SISTEMA MULTI-AGENTE DE LICITACIONES.....	43
3.1.1. LICITACION .....	43

3.1.2. SISTEMA MULTI-AGENTE DE LICITACIONES .....	43
3.1.3. NEGOCIACION.....	44
3.1.3.1 PROPUESTAS POSIBLES DE LA NEGOCIACION.....	44
3.1.3.2. PROTOCOLO DE NEGOCIACION .....	44
3.1.3.3. ESTRATEGIA DE NEGOCIACION.....	44
3.2. SMA DE LICITACIONES CON MAS-COMMONKADS.....	45
3.2.1. CONCEPTUALIZACION .....	45
3.2.2. ANALISIS .....	47
3.2.2.1. MODELO DE AGENTES .....	47
3.2.2.2. MODELO DE TAREAS .....	49
3.2.2.3. MODELO DE COORDINACIÓN .....	51
3.2.2.4. MODELO DE EXPERIENCIA .....	53
3.2.2.5. MODELO DE ORGANIZACIÓN.....	54
3.2.3. DISEÑO.....	55
3.2.3.1. MODELO DE COMUNICACIÓN.....	55
3.2.3.2. EL MODELO DE RED.....	55
3.3. CASO DE ESTUDIO LICITACIONES CON INGENIAS.....	56
3.3.1. IMPLEMENTACION EN IDK.....	57
3.3.1.1 MODELO DE AGENTE.....	57
3.3.1.2. MODELO DE OBJETIVOS Y TAREAS .....	58
3.3.1.3. MODELO DE INTERACCION.....	60
3.3.1.4. MODELO DE ORGANIZACION .....	62
3.3.1.5. MODELO DE ENTORNO.....	63
3.4. CASO DE ESTUDIO LICITACIONES IMPLEMENTADO EN JADE.....	65
<b>CAPITULO IV. EVALUACION Y COMPARACION DE METODOLOGIAS .....</b>	<b>69</b>
4.1. CRITERIOS EVALUATIVOS Y COMPARATIVOS .....	69
4.1.1. LA BASE DE EVALUACION .....	70
4.1.2. CRITERIOS RELACIONADOS AL PROCESO .....	70
4.1.3. CRITERIOS RELACIONADOS A LA TECNICA .....	72
4.1.4. CRITERIOS RELACIONADOS AL MODELO .....	72
4.1.5. CRITERIOS CARACTERISTICA ALENTADORA.....	74
4.2. ANÁLISIS COMPARATIVO .....	75
4.2.1. COMPARACION DE LOS CRITERIOS RELACIONADOS AL MODELO .....	75
4.2.2. COMPARACION DE CRITERIOS RELACIONADOS A LA TECNICA.....	76
4.2.3. COMPARACION DE CRITERIOS RELACIONADOS AL MODELO .....	79
4.2.4. COMPARACION DE CRITERIOS CARACTERISTICA ALENTADORA .....	81

<b>CAPITULO V. CONCLUSIONES</b> .....	82
5.1. INTRODUCCION .....	82
5.2 CONCLUSIONES GENERALES .....	83
5.3. RECOMENDACIONES .....	84
<b>REFERENCIAS BIBLIOGRAFICAS</b> .....	85
<b>ANEXOS</b> .....	87
ANEXO A TABLAS DE COMPARACION .....	88
ANEXO B PRUEBAS DE PROTOTIPO DE SMA DE LICITACIONES.....	91
<b>DOCUMENTACION</b> .....	97

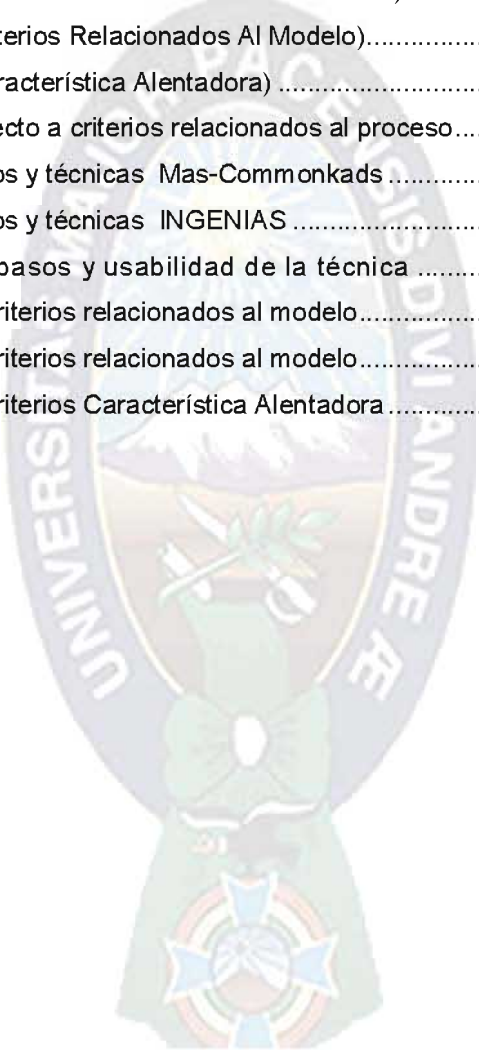




## INDICE DE TABLAS

<b>Tabla 1.</b> Documento 1 (Características de MAS-CommonKADS) .....	17
<b>Tabla 2.</b> Documento 2 (Fases de Desarrollo de MAS-CommonKADS) .....	18
<b>Tabla 3.</b> Documento 3 (Modelos de MAS-CommonKADS).....	19
<b>Tabla 4.</b> Documento 4 (Características de INGENIAS).....	28
<b>Tabla 5.</b> Documento 5 (Componentes Notacionales de INGENIAS) .....	29
<b>Tabla 6.</b> Documento 6 (Meta-Modelos de INGENIAS).....	31
<b>Tabla 7.</b> Documento 7 (IDK) .....	38
<b>Tabla 8.</b> Documento 8 (Fase de Conceptualización).....	45
<b>Tabla 8.1.</b> Descripción de Caso de Uso- Licitar Artículo .....	46
<b>Tabla 8.2.</b> Descripción de Caso de Uso-Buscar Artículos.....	46
<b>Tabla 8.3.</b> Descripción de Caso de Uso-Realizar Oferta.....	46
<b>Tabla 8.4.</b> Descripción de Caso de Uso-Anunciar Ganador.....	47
<b>Tabla 9.</b> Documento 9 (Modelo de Agente).....	47
<b>Tabla 9.1.</b> Tarjeta CRC para Agente Licitante,.....	47
<b>Tabla 9.2.</b> Tarjeta CRC para Agente Ofertante .....	48
<b>Tabla 10.</b> Documento 10 (Modelo de Tareas) .....	49
<b>Tabla 10.1.</b> Descripción Tarea Publicar Licitación.....	49
<b>Tabla 10.2.</b> Descripción Tarea Realizar Oferta .....	50
<b>Tabla 10.3.</b> Descripción Tarea Decidir Ganador .....	50
<b>Tabla 10.4.</b> Descripción Tarea Minimizar Precio.....	50
<b>Tabla 10.5.</b> Descripción Tarea Ofertar Bajo .....	50
<b>Tabla 10.6.</b> Descripción Tarea Realizar Oferta .....	50
<b>Tabla 10.7.</b> Descripción Tarea Realizar Oferta .....	51
<b>Tabla 11.</b> Documento 11 (Modelo de Coordinación).....	51
<b>Tabla 11.1.</b> Descripción Conversación Licitar Artículo .....	52
<b>Tabla 11.2.</b> Descripción Conversación Oferta .....	52
<b>Tabla 11.3.</b> Descripción Conversación Oferta Ganadora.....	52
<b>Tabla 11.4.</b> Descripción De Grupo de Agentes Ofertantes .....	53
<b>Tabla 12.</b> Documento 12 (Modelo de Experiencia) .....	53
<b>Tabla 12.1.</b> Cuadro de Experiencia .....	54
<b>Tabla 13.</b> Documento 13 (Modelo de Organización).....	54
<b>Tabla 14.</b> Documento 14 (Diseño de MAS-CommonKADS) .....	55
<b>Tabla 15.</b> Documento 15 (Notación de Casos de Uso de INGENIAS).....	56

<b>Tabla 16.</b> Documento 16 (Especificación de diagramas de INGENIAS).....	57
<b>Tabla 17.</b> Documento 17 (Modelo de Agente de INGENIAS) .....	57
<b>Tabla 18.</b> Documento 18 (Modelo de Objetivos y Tareas de INGENIAS).....	59
<b>Tabla 19.</b> Documento 19 (Modelo de Interacciones de INGENIAS) .....	60
<b>Tabla 20.</b> Documento 20 (Modelo de Organización de INGENIAS).....	62
<b>Tabla 21.</b> Documento 21 (Modelo de Entorno de INGENIAS) .....	64
<b>Tabla 22.</b> Documento 22 Criterios Relacionados Al Proceso.....	70
<b>Tabla 23.</b> Documento 23 (Criterios Relacionados A La Técnica).....	72
<b>Tabla 24.</b> Documento 24 (Criterios Relacionados Al Modelo).....	72
<b>Tabla 25.</b> Documento 25 (Característica Alentadora) .....	74
<b>Tabla 26.</b> Comparación respecto a criterios relacionados al proceso.....	75
<b>Tabla 27.</b> Evaluación de pasos y técnicas Mas-Commonkads .....	77
<b>Tabla 28.</b> Evaluación de pasos y técnicas INGENIAS .....	78
<b>Tabla 29.</b> Comparación de pasos y usabilidad de la técnica .....	79
<b>Tabla 30.</b> Comparación de criterios relacionados al modelo.....	79
<b>Tabla 31.</b> Comparación de criterios relacionados al modelo.....	80
<b>Tabla 32.</b> Comparación de criterios Característica Alentadora .....	81



## INDICE DE FIGURAS

<b>FIGURA 2.1.</b> Descripción del agente y su entorno.....	9
<b>FIGURA 2.2.</b> Descripción del agente.....	10
<b>FIGURA 2.3.</b> Arquitectura Abstracta de FIPA.....	12
<b>FIGURA 2.4.</b> Descripción del agente.....	14
<b>FIGURA 2.5.</b> Los modelos de MAS-CommonKADS .....	19
<b>FIGURA 2.6.</b> Tarjetas CRC para Agentes .....	20
<b>FIGURA 2.7.</b> Notación alternativa para representar la descomposición de tareas .....	21
<b>FIGURA 2.8.</b> Principales componentes del modelo de la experiencia .....	22
<b>FIGURA 2.9.</b> Descomposición del modelo de diseño.....	25
<b>FIGURA 2.10.</b> Constituyentes del diseño de red .....	25
<b>FIGURA 2.11.</b> Diseño de la plataforma .....	26
<b>FIGURA 2.12.</b> Entidades básicas de la metodología .....	29
<b>FIGURA 2.13.</b> Relaciones entre los diferentes meta-modelos y las dos entidades principales.....	31
<b>FIGURA 2.14.</b> Diagrama de descripción de Agente .....	32
<b>FIGURA 2.15.</b> Diagrama del modelo de agente .....	32
<b>FIGURA 2.16.</b> Elementos del modelo de agente.....	33
<b>FIGURA 2.17.</b> Descomposición de objetivos-Árboles Y/O .....	34
<b>FIGURA 2.18.</b> Tareas asociadas a Objetivos.....	34
<b>FIGURA 2.19.</b> Elementos de definición de tareas .....	34
<b>FIGURA 2.20.</b> Descripción estructural.....	35
<b>FIGURA 2.21.</b> Descripción social .....	36
<b>FIGURA 2.22.</b> Descripción funcional .....	36
<b>FIGURA 2.23.</b> Naturaleza de la interacción.....	37
<b>FIGURA 2.24.</b> Diagrama Ingenias Development Kit (IDK) .....	38
<b>FIGURA 2.25.</b> Generación de código de IDK.....	39
<b>FIGURA 2.26.</b> Generación del editor .....	39
<b>FIGURA 2.27.</b> Arquitectura de JADE.....	41
<b>FIGURA 3.1.</b> Casos de Uso del SMA de Licitaciones .....	46
<b>FIGURA 3.2.</b> Tareas del SMA de Licitaciones.....	49
<b>FIGURA 3.3.</b> Conversación en Licitación .....	51
<b>FIGURA 3.4.</b> Canales De Comunicación.....	53
<b>FIGURA 3.5.</b> Caso reactivo para Comprar .....	54
<b>FIGURA 3.6.</b> Caso reactivo para Vender.....	55

<b>FIGURA 3.7.</b> Caso de Uso para INGENIAS .....	56
<b>FIGURA 3.8.</b> Especificación de diagramas.....	57
<b>FIGURA 3.9.</b> Diagrama del Modelo de Agente.....	57
<b>FIGURA 3.10.</b> Entidades del Modelo de Agente .....	58
<b>FIGURA 3.11.</b> Relaciones que aparecen en el Modelo de Agente .....	58
<b>FIGURA 3.12.</b> Diagrama del Modelo de Objetivos y tareas .....	59
<b>FIGURA 3.13.</b> Entidades que aparecen en el Modelo de Objetivos y Tareas .....	60
<b>FIGURA 3.14.</b> Relaciones que aparecen en el Modelo de Objetivos y Tareas.....	60
<b>FIGURA 3.15.</b> Diagrama del Modelo de Interacción .....	61
<b>FIGURA 3.16.</b> Entidades que aparecen en el Modelo de Interacción .....	61
<b>FIGURA 3.17.</b> Relaciones que aparecen en el Modelo de Interacción .....	61
<b>FIGURA 3.18.</b> Diagrama del Flujo de Evento del Modelo de Organización.....	62
<b>FIGURA 3.19.</b> Entidades que aparecen en el Flujo de Eventos.....	63
<b>FIGURA 3.20.</b> Relaciones que aparecen en el Flujo de Evento .....	63
<b>FIGURA 3.21.</b> Especificación del Ambiente del Modelo.....	64
<b>FIGURA 3.22.</b> Entidades que aparecen en el Ambiente del Modelo .....	64
<b>FIGURA 3.23.</b> Relaciones que aparecen en el Ambiente del Modelo.....	64
<b>FIGURA 3.24.</b> Proyecto Licitación .....	65
<b>FIGURA 3.25.</b> Clases del Proyecto Licitación .....	66
<b>FIGURA 3.26.</b> Configuración de Parámetros de entrada .....	67
<b>FIGURA 3.27.</b> Clases del Proyecto Licitación .....	67
<b>FIGURA 3.28.</b> Pizarra de Licitación .....	68
<b>FIGURA 3.29.</b> Consola de Eclipse en la Ejecución de Licitación.....	68
<b>FIGURA 4.1.</b> Estructura de la base de evaluación .....	70

# **CAPITULO I**

## **MARCO REFERENCIAL**

### **1.1. INTRODUCCION**

En la última década y media la tecnología de agentes recibió una mayor atención en el desarrollo investigativo científico, en consecuencia las empresas e industrias empezaron a interesarse en la tecnología de Agentes para desarrollar sus propios productos.

Hoy aunque ya se han desarrollado numerosos agentes y sistemas multi-agentes, y a pesar de que se continúa discutiendo qué es un agente, la respuesta mas sencilla es que básicamente un agente es una entidad de software que exhibe un comportamiento autónomo y un sistema multi-agente es un conjunto de agentes que tienen la capacidad de interactuar en un entorno común. Los Sistemas Multi Agentes-SMA surgieron como una nueva área de investigación en los años 90, desarrollándose desde los campos del procesamiento distribuido y la inteligencia artificial.

El desarrollo de sistemas multi-agentes es hoy en día uno de los temas de mayor discusión en el área de Agentes. En muchas aplicaciones es necesario tener sistemas que puedan decidir por ellos mismos que necesitan para satisfacer sus objetivos y presentar una solución a un usuario. Tales sistemas son conocidos como agentes (agentes inteligentes o agentes autónomos), los sistemas tienen que actuar en entornos cambiantes e impredecibles, donde existe la posibilidad de que las acciones emprendidas puedan fallar.

El paradigma de agentes y sistemas multi-agente constituye actualmente un área de creciente interés dentro de la Inteligencia Artificial. Numerosas aplicaciones basadas en el nuevo paradigma vienen ya siendo empleadas en infinidad de áreas, debido a la

utilización de los sistemas multi-agente para el desarrollo de aplicaciones complejas. Cada vez son más necesarios métodos, técnicas y herramientas que faciliten el desarrollo de aplicaciones basadas en dicho paradigma.

El progreso en la ingeniería del software en los últimos años se ha realizado gracias al desarrollo de abstracciones más poderosas y naturales para modelar sistemas más complejos. Si consideramos a los agentes con el suficiente potencial como un paradigma de la ingeniería del software, entonces es necesario desarrollar técnicas de ingeniería del software que sean específicamente aplicables a este paradigma. La introducción de la tecnología de agentes en la industria requiere de metodologías que asistan en todas las fases del ciclo de vida del sistema de agentes.

En los últimos años han aparecido diferentes aproximaciones que tratan de presentar una metodología apropiada para el desarrollo de sistemas multi-agente.

En el presente trabajo de tesis, en primera instancia se presenta, una visión generalizada de dos propuestas, las más representativas que son MAS-commonKads e INGENIAS, la metodología MAS-commonKads basada en el conocimiento y experiencia, e INGENIAS que orientada a Agentes. Las propuestas son analizadas desde un punto de vista conceptual, y ambas metodologías son aplicadas a un caso de estudio, así al modelar un mismo Sistema Multi-Agente se presentan en forma detallada los pasos que se deben seguir en cada una de ellas para llegar desde la especificación del problema a una solución implementable con ambas metodologías.

## **1.2. ANTECEDENTES**

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

La solución de problemas complejos no se lleva a cabo con un único agente, sino que es necesario un conjunto de agentes que interactúan entre ellos para la consecución de los objetivos del sistema. La utilización de varios agentes representa de forma natural la

descentralización del sistema y enfatiza la comunicación para gestionar las dependencias entre ellos **[PAR05]**.

Un Sistema Multi-Agente -SMA consiste en un número de agentes que interactúan entre sí a través de la comunicación, para interactuar exitosamente los agentes deben ser capaces de cooperar, coordinarse, y negociar con otros, tal como lo hacen los humanos. Es decir, los agentes tienen un grado de autonomía mayor y pueden decidir dinámicamente qué interacciones son adecuadas, qué tareas deben realizar, quién realiza cada tarea y, además, es posible mantener conocimiento que no es globalmente consistente, incluso los agentes pueden mantener objetivos globales diferentes. En un caso general, los agentes actúan en nombre de los usuarios, con diferentes objetivos y motivaciones.

La introducción de la tecnología de agentes en la industria requiere de metodologías que asistan en todas las fases del ciclo de vida del sistema de agentes. El paradigma de agentes y sistemas multi-agente constituye actualmente un área de creciente interés dentro de la Inteligencia Artificial. Numerosas aplicaciones basadas en este nuevo paradigma vienen ya siendo empleadas en infinidad de áreas, tales como control de procesos, manipulación y filtrado de información aplicados a diferentes dominios, agentes bancarios, ventas y comercio electrónico, difusión de noticias y publicidad, realidad virtual y avatares, control de procesos y manufactura, telecomunicaciones, juegos entre otros **[MAR00]**. Los SMA tienen un amplio rango de aplicaciones, uno de los campos principales la constituyen los agentes software, los cuales se focalizan principalmente en la explotación cooperativa de recursos accesibles por Internet. Los SMA tienen un gran potencial de aplicación para la solución de problemas de la vida práctica y por ende para el desarrollo de nuevos productos.

Cuando se dieron los primeros pasos en un entorno distribuido de la Inteligencia Artificial, se busco establecer métodos o normativas que ayuden al desarrollo de SMA's En la actualidad cobra mucha fuerza el hecho de contar con una metodología que ayude en gran medida al desarrollo de Sistemas Multi-Agentes (SMA) **[FMS97]**, **[JEJ95]**, **[JEJ95]**, **[MUL96]**, los cuales integran a su desarrollo aspectos de ingeniería de software clásicos, lo cual permite evidenciar de manera objetiva la completitud de fases de desarrollo, consiguiendo así un grado de certidumbre con respecto a la calidad de sistemas.

Una metodología constituye un conjunto de etapas o fases que van guiando a los diseñadores partiendo de una definición de alto nivel hacia una aproximación a una implementación. Si estas etapas son pocas entonces es demasiado general y el salto que existe entre una y otra es muy grande. En el otro extremo están las metodologías con muchas etapas y una granularidad muy fina que puede ser tediosa de usar por aquellas personas con alguna experiencia.

Por otra parte, la aceptación de métodos y metodologías en la industria y/o empresa depende de la existencia de las herramientas necesarias que soporten el análisis, diseño e implementación de agentes software. Con el desarrollo de teorías, arquitecturas y lenguajes de agentes en estudios realizados en torno a las metodologías orientadas a Agentes se optó por filtrar la investigación entre tres y siete las metodologías más representativas y completas que existen en la actualidad, algunas de las temáticas investigadas son las siguientes:

- *Metodología para el desarrollo de Sistemas Multi-Agente [PAV04]* es una línea de trabajo que consolida experiencia adquirida en forma de metodología, presenta un resumen de lo que proponen las metodologías actuales, el autor selecciona siete metodologías relevantes ante la imposibilidad de revisar todas.
- *Metodologías de desarrollo de sistemas multi-agente: Un análisis comparativo [MAR00]* Este trabajo presenta y compara algunas metodologías de desarrollo de Sistemas Multi-Agente, y tiene como objetivo extraer cuales son los principios básicos que deberían contemplarse al desarrollar una plataforma o una metodología para este tipo de sistemas.
- *Definición de una metodología para el desarrollo de Sistemas Multi-Agente [IGL98]* Esta tesis define una metodología para el desarrollo de Sistemas Multi-Agente, integrando técnicas de ingeniería del conocimiento, ingeniería software orientada a objetos e ingeniería software de protocolos.

Y finalmente al revisar trabajos previos acerca de metodologías de desarrollo de Sistemas Multi-Agente, en la Universidad Mayor de San Andrés se hizo una investigación sobre Estandarización de metodologías de SMA's, como la del autor Richard Rubén Velasco Ali con el título: "Estandarización de Metodologías para el desarrollo de Sistemas Multi-Agente". Esta tesis prueba de una manera efectiva el desarrollo de agentes con varias metodologías, dando al lector una manera factible de crear un agente.



### **1.3. PLANTEAMIENTO DEL PROBLEMA**

El desarrollo de Sistemas Multi-Agente tomo gran importancia en los últimos años, por ser aplicable a la resolución de problemas complejos no resueltos de manera satisfactoria mediante técnicas clásicas. Al mismo tiempo de llegar a ser una tecnología con varios beneficios creo nuevas necesidades de desarrollo, es decir requerimientos de usuarios para la construcción de SMA's.

La introducción de la tecnología de agentes en la industria requiere de metodologías que asistan en todas las fases del ciclo de vida del sistema de agentes. Actualmente existen varias metodologías de construcción de SMA's, cada metodología, se especializa en áreas concretas [MAR00], como se menciono antes, así cada una de las metodologías de desarrollo de SMA's se encuentran debilitadas o fortalecidas en algunas fases, entonces surge el dilema de elegir una metodología adecuado para desarrollar un SMA.

Por lo tanto en base a lo mencionado anteriormente se puede plantear la siguiente pregunta:

**¿Es posible realizar una comparación y evaluación de las metodologías Mas-CommonKADS e INGENIAS y responder a las necesidades particulares de los desarrolladores en cuanto a las practicas de desarrollo de Sistemas Multi-Agente?**

### **1.4. OBJETIVOS DEL PROYECTO**

#### **1.4.1. OBJETIVO GENERAL**

Establecer un comparación y evaluación de las metodologías MAS-commonKads e INGENIAS aplicadas a un caso de estudio en especifico un Sistema Multi-Agente de Licitaciones, para mejorar las practicas en cuanto al desarrollo de Sistemas Multi-Agente.

#### **1.4.2. OBJETIVOS ESPECIFICOS**

Para lograr el objetivo general a continuación se mencionan los objetivos específicos por módulos:

- ❖ Modelar un Sistema Multi-Agente de licitaciones usando las metodologías MAS-CommonKads e INGENIAS para formular un comparativo de metodologías en la construcción de SMA en las fases de Análisis, Diseño y Desarrollo.

- ❖ Proporcionar recomendaciones para el desarrollo de agentes usando las metodologías MAS-commonKads e INGENIAS, a través del modelado del SMA de licitaciones con cada metodología.
- ❖ Implementar agentes independientes para crear el prototipo de la tesis mediante el desarrollo de agentes en una plataforma JADE.

## 1.5. HIPOTESIS

La hipótesis de trabajo es:

Al modelar un Sistema Multi-Agente de Licitaciones con una metodología basada en Conocimiento como MAS-Commonkads los resultados obtenidos en cuanto a los criterios relacionados al Proceso, la Técnica, el Modelo y la Característica Alentadora de la metodología deberían ser significativamente iguales a los obtenidos al modelar el mismo Sistema Multi-Agente con una metodología orientada a Agentes como Ingenias.

## 1.6. JUSTIFICACION

### 1.6.1. JUSTIFICACION CIENTIFICA

El presente trabajo de tesis se basa en resolver el dilema existente de elegir una metodología. Las metodologías proporcionan medios para construir Sistemas Multi-Agente de forma disciplinada y repetible, cada metodología, por el bagaje de sus creadores, se especializa en áreas concretas.

La selección atiende a la presencia de un proceso de desarrollo, el combinar diferentes vistas para describir el sistema, e incorporar elementos asociados al área de los agentes.

### 1.6.2. JUSTIFICACION TECNICA

En muchos casos se utilizan varios métodos de desarrollo y posterior evaluación del SMA, apoyándose en los resultados de la investigación en agentes y Sistemas Multi-Agente, surge una línea de trabajo cuyo propósito es consolidar la experiencia adquirida en forma de metodologías, en el actual caso de dos metodologías. Para conseguir el propósito trazado en cuanto a aspectos técnicos la utilización de equipos de cómputo a partir de Pentium IV son necesarios, debido a la plataforma a utilizar para realizar el prototipo que requiere de memoria y velocidad en el equipo de cómputo.

### 1.6.3. JUSTIFICACION SOCIAL

Se justifica socialmente puesto que este trabajo de investigación será reflejado en beneficio directo del desarrollador que opta por agentes múltiples, el desarrollador podrá elegir el método de desarrollo mas útil a su necesidad, al observar que cada metodología fue evaluada en un SMA en concreto.

### 1.7. METODOLOGIA

Para demostrar la hipótesis que es fundamento principal de este trabajo de tesis se identifican métodos y técnicas generales y específicas.

Los métodos específicos que aportan con una visión mas concreta relacionado a los temas de desarrollo son:

- ❖ Metodología de desarrollo Orientado a Agentes:
  - Basado en metodología orientado a Objetos
  - Basado en metodología de ingeniería de conocimiento Kads.
- ❖ Metodología de desarrollo de SMA (MAS-CommonKADS E INGENIAS)

Los métodos generales que ayudan ala demostración de la hipótesis en un marco conceptual son:

- ❖ Comparativo evaluativo de Metodologías
- ❖ Método Científico: el cual tiene los siguientes pasos para llegar a probar la tesis
  - *Observación*: Observar es aplicar atentamente los sentidos a un objeto de estudio, para estudiarlos tal como se presentan en realidad, puede ser ocasional o causalmente.
  - *Inducción*: La acción y efecto de extraer, a partir de determinadas observaciones o experiencias particulares, el principio particular de cada una de ellas.
  - *Hipótesis*: Planteamiento mediante la observación siguiendo las normas establecidas por el método científico.
  - Probar la hipótesis por *experimentación*.
  - Tesis o teoría científica (*conclusiones*).

## **1.8. ALCANCES Y APORTES**

### **1.8.1. LIMITES Y ALCANCES**

El presente trabajo toma dos de las más conocidas metodologías de desarrollo de Sistemas Multi-Agente y realiza una comparación de ambas metodologías pero esta vez de manera práctica aplicada a un caso de estudio, modelando el mismo SMA con ambas metodologías.

Al modelar un mismo Sistemas Multi-Agente en este caso un SMA de licitaciones con ambas metodologías obtendremos resultados parecidos o no, respecto de cada metodología lo cual podría ayudar al modelador del SMA a decidir con cual de las dos metodologías trabajar acorde a su necesidad de acuerdo a las características antes mencionadas en Antecedentes.

### **1.8.2 APORTES**

El aporte de este trabajo de investigación es realizar un análisis comparativo de las metodologías INGENIAS y MAS-COMMOMKADS en base a las responsabilidades y el comportamiento del proceso interno de desarrollo y del proceso de software de un mismo sistema multi-agente es posible para obtener un cuadro comparativo con datos estructurados y no estructurados, para que el desarrollador seleccione bajo criterios cualitativos una metodología acorde a sus necesidades

## CAPITULO II

### MARCO CONCEPTUAL

#### 2.1. AGENTES

Un agente es una entidad física o abstracta que puede percibir su ambiente a través de sensores, es capaz de evaluar percepciones y tomar decisiones por medio de mecanismos de razonamiento sencillos o complejos, comunicarse con otros agentes para obtener información y actuar sobre el medio en el que se desenvuelve a través de ejecutores [MAR00].

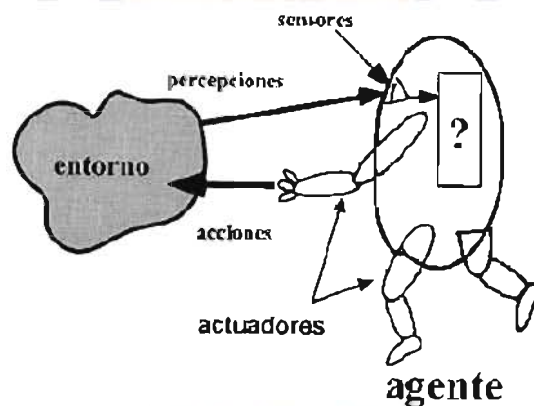


FIGURA 2.1. Descripción del agente y su entorno  
Fuente [MAR00]

Un agente puede ser clasificado por su rol o papel en una sociedad de agentes (SMA al que pertenece), por su especialidad o actividad en la que es experto, por sus objetivos o metas en el SMA, por su funcionalidad o lo que es capaz de hacer, por sus creencias, por su capacidad de comunicación y por su capacidad de aprendizaje.

Así que se identifican las siguientes clases de agentes [ROD07]:

- ♦ **Agentes Colaborativos**, son capaces de actuar racionalmente y autónomamente en ambientes Multi-Agente y con restricciones de recursos. Otras características

de los agentes es que poseen habilidades sociales, son pro-activos, benévolos, estáticos y veraces.

- ◆ **Agentes de Interfase**, enfatizan en la autonomía y la adaptabilidad para realizar tareas a sus usuarios. Los agentes de interfaces básicamente presta soporte y asistencia a un usuario que está aprendiendo una nueva aplicación o nuevos conceptos.
- ◆ **Agentes móviles**, enfatizan en las habilidades sociales y la autonomía, a diferencia de los agentes cooperativos, son móviles.
- ◆ **Agentes de información**, nacieron debido a la gran cantidad de herramientas que surgieron para el manejo y recuperación de información. Los agentes de información tienen los roles de manejar, manipular, e integrar información de muchas fuentes de datos distribuidas.

### 2.1.1 AGENTES INTELIGENTES

Un agente inteligente es un sistema informático, situado en un entorno, que es capaz de realizar acciones flexibles y autónomas para alcanzar sus objetivos” [JEW95]. Lo anterior no quiere decir que el agente tenga dominio completo del ambiente, por el contrario en la mayoría de situaciones el agente es muy complejo y solo tendrá un control parcial del ambiente. Así mismo puede suceder que para un estado del ambiente muy similar, dos acciones diferentes tomadas por el agente produzcan efectos heterogéneos en el ambiente.

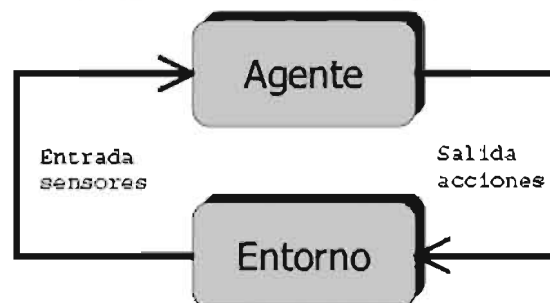


FIGURA 2.2. Descripción del agente  
Fuente. [STA07]

Realizan acciones flexibles para conseguir sus **objetivos** como:

- ◆ **Reactivo**, el agente es capaz de responder a cambios en el entorno en que se encuentra situado.
- ◆ **Pro-activo**, a su vez el agente debe ser capaz de intentar cumplir sus propios planes u objetivos.

- ◆ **Social**, debe poder comunicarse con otros agentes mediante algún tipo de lenguaje de comunicación de agentes. Interacción con otros agentes como: Cooperación, competencia, negociación.

A grandes rasgos, las **características** típicas de un agente inteligente son **[PAV04]**:

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

## 2.2. SISTEMAS MULTI-AGENTE

Los **Sistemas Multi-Agente** son sistemas distribuidos de software formados por un conjunto de agentes autónomos (nodos) que trabajan juntos para resolver problemas. Asignándose tareas unos a otros y colaborando, los agentes pueden solucionar problemas que serían irresolubles para un agente aislado **[ROD07]**. En un Sistema Multi-Agente no hay un control global del sistema ni un lugar donde esté toda la información. La "inteligencia" de un SMA puede obtenerse de dos maneras. En primer lugar, mediante el uso de agentes inteligentes para el sistema. En segundo lugar, usando agentes reactivos

(son aquellos que reaccionan siguiendo el esquema conductista “estímulo-respuesta”). En este caso, la inteligencia colectiva del sistema es un fenómeno emergente.

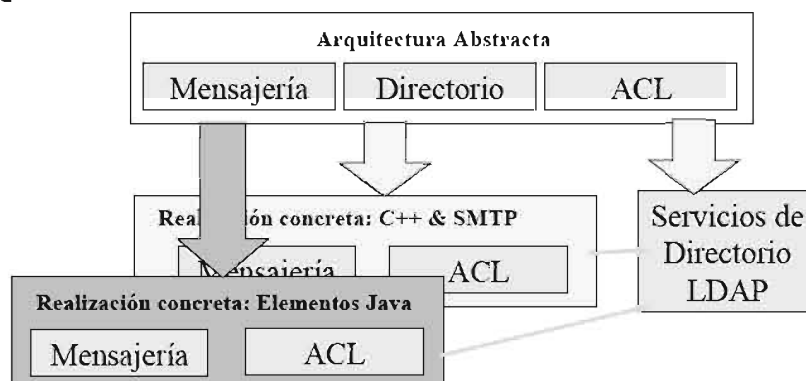
La construcción de SMA integra tecnologías de distintas áreas de conocimiento: Técnicas de ingeniería del software para estructurar el proceso de desarrollo; técnicas de inteligencia artificial para dotar a los programas de capacidad para tratar situaciones imprevistas y tomar decisiones, y programación concurrente y distribuida para tratar la coordinación de tareas ejecutadas en diferentes máquinas bajo diferentes políticas de planificación.

### 2.2.1. FIPA (**FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS**)

Un consorcio industrial fundado en 1996, reunidas varias decenas de compañías de telecomunicaciones e Informática, sus objetivos:

- ◆ Acelerar el desarrollo de tecnologías de agentes inteligentes mediante la producción de especificaciones acordadas internacionalmente
- ◆ Especificación del comportamiento y capacidades externas de subsistemas genéricos, recursos de agentes (para migración, ejecución, etc.), interacción y cognitivos (las especificaciones estarán basadas en casos prácticos concretos)
- ◆ Agentes, multi-agentes, y sociedades de agentes.
- ◆ Selección y adaptación de tecnologías existentes.

### 2.2.2. ARQUITECTURA ABSTRACTA DE FIPA



**FIGURA 2.3.** Arquitectura Abstracta de FIPA  
Fuente. [PAV05]

Los agentes se comunican intercambiando mensajes que representan actos de habla codificados en un lenguaje de comunicación de agentes

Los servicios de soporte a los agentes son servicios de directorio y servicios de transporte de mensajes. Los servicios se pueden implementar como agentes, como software que se accede invocando métodos: C++, Java.



### 2.2.3. FIPA ACL (*Agent Communication Language*)

Basado en actos del habla, la semántica se basa en actitudes mentales (creencias, intenciones, etc.) Los mensajes son acciones comunicativas que tienen una sintaxis similar a KQML<sup>1</sup>, semántica formal definida con lógica modal, además se definen protocolos de interacción de alto nivel, llamados conversaciones.

Es posible definir nuevas primitivas a partir de un núcleo de primitivas mediante composición, el estado de los agentes se describe con el lenguaje SL (*Semantic Language*) que es una lógica multi-modal con operadores modales para creencias (B), deseos (D), creencias falsas (U), e intenciones u objetivos persistentes (PG), con SL se pueden representar proposiciones, objetos y acciones.

ACL se define como un conjunto de fórmulas SL que describen:

- ◆ Las precondiciones de factibilidad (FP), las condiciones necesarias para el emisor del acto comunicativo aunque se den las condiciones el agente no está obligado a realizar el acto comunicativo.
- ◆ El efecto racional (RE), lo que un agente espera que ocurra como resultado de realizar la acción pero el agente receptor no está obligado a asegurar que ocurra el efecto esperado ya que incluso puede resultarle imposible lograrlo.

En comparación con KQML sintácticamente son muy similares, ningún lenguaje se compromete con un solo lenguaje de contenidos, se diferencian básicamente en la interpretación semántica de las preformativas

En la descripción KQML utiliza pre- y post- condiciones y FIPA ACL usa precondiciones de factibilidad (FP, *feasability preconditions*) y efecto racional (RE, *rational effect*). FIPA ACL es más potente en la composición de nuevas primitivas.

### 2.2.4. COMUNICACION ENTRE AGENTES

La comunicación es la base para las interacciones y la organización social de los agentes. Hay interacciones cuando la dinámica de un agente está perturbada por las influencias de otros, las interacciones son el motor de los SMA., existen distintas formas de interaccionar:

- ◆ Acciones sobre el entorno
- ◆ Pizarra compartida
- ◆ Inferencias

---

1. *Knowledge Query and Manipulation Language*, es un lenguaje de comunicación y protocolo, orientado a mensajes, para el intercambio de información. KQML es independiente de protocolos de transporte (TCP/IP, HTTP,...), de sintaxis de contexto ontologías y de protocolos de alto nivel.

◆ Paso de mensajes

a) **Comunicación Mediante el Entorno**, se basa en la “**Arquitectura de sub-sunción cooperativa**”, la arquitectura de sub-sunción es una cooperación que no usa comunicación directa, la comunicación se realiza a través del entorno. Hay dos comportamientos que se ejecutan en paralelo en la comunicación mediante el entorno:

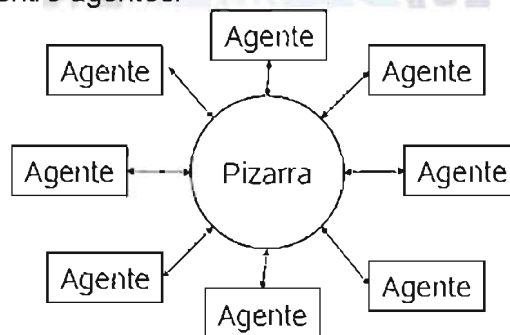
◆ **Comportamientos de manejo de objetos**

Ej.: Si detecto una piedra y no llevo ninguna -> recogerla.

◆ **Comportamientos de movimiento**

Organizados de acuerdo a una jerarquía de sub-sunción.

b) **Sistemas De Pizarra**, la Pizarra es la zona de trabajo común que permite a los agentes compartir todo tipo de información. Un sistema multi-agente puede tener varias pizarras con distintos agentes registrados en cada una, lo cierto es que no hay comunicación directa entre agentes.



**FIGURA 2.4.** Descripción del agente  
Fuente. [PAV05]

Los sistemas más avanzados incorporan nuevos conceptos como un:

◆ **Moderador**, que es el Agente especializado con conocimiento de control y de evaluación que publica en la pizarra los sub-problemas a resolver y decide a qué agentes se asignan de entre aquellos que se han ofrecido a resolverlos.

◆ **Despachador**, agente que avisa a los agentes afectados por algún cambio producido en la pizarra

El método flexible de comunicación para la resolución distribuida de problemas son independientes de la estrategia de cooperación que se va a utilizar y no afectan a la arquitectura de los agentes individuales.

Sin embargo, la estructura central de la pizarra representa cada vez más un inconveniente ya que todos los agentes distribuidos por una red se ven obligados a acceder al dispositivo central donde se encuentra la pizarra.

## 2.3. METODOLOGIAS PARA EL DESARROLLO DE SMA's

Para desarrollar sistemas basados en agentes, se requiere de una metodología que apoye los procesos de desarrollo comunes.

Una metodología es un conjunto de guías para cubrir todo el ciclo de desarrollo de un sistema, y debe proveer:

- ◆ El ciclo de vida del proceso completo
- ◆ Un conjunto de conceptos y modelos
- ◆ Un conjunto de técnicas
- ◆ Un conjunto de métricas
- ◆ Aseguramiento de la calidad
- ◆ Estándares de codificación
- ◆ Consejos para la reutilización, y
- ◆ Guías para la administración de proyectos

Existen más de 20 metodologías distintas para el desarrollo de Sistemas Multi-Agentes.

Ejemplos:

- |                  |               |
|------------------|---------------|
| ◆ GAIA           | ◆ Adept       |
| ◆ MAS-CommonKADS | ◆ AUML        |
| ◆ Tropos         | ◆ MASSIVE     |
| ◆ MaSE           | ◆ MESSAGE/UML |
| ◆ Cassiopeia     | ◆ PASSI       |
| ◆ AALADIN        | ◆ INGENIAS    |
| ◆ DESIRE         |               |

Son tres enfoques distintos usados en el desarrollo de dichas metodologías:

- ◆ **El enfoque orientado a objetos**, son metodologías que han evolucionado de tecnologías orientadas a objetos, como: AUML, MESSAGE/UML, INGENIAS.
- ◆ **El enfoque de ingeniería del conocimiento**, son metodologías que han evolucionado de la ingeniería del conocimiento, su énfasis es en la identificación, adquisición, y modelado del conocimiento que usará el agente. Como: CoMoMAS, MAS-CommonKADS, TROPOS.
- ◆ **El enfoque de teoría de agentes**, son metodologías derivadas de la teoría de agentes, y se caracterizan por su enfoque en capturar abstracciones a nivel de socialización. Tales como: KAoS, Vowel Engineering, Gaia.

## **2.4. METODOLOGIA MAS-COMMONKADS**

### **2.4.1. PERSPECTIVA HISTORICA MAS-COMMONKADS**

CommonKADS es una metodología diseñada para el desarrollo de Sistemas Basados en conocimiento-SBC de forma análoga a los métodos empleados en ingeniería software. El desarrollo de la metodología ha sido financiado por la Comunidad Europea entre 1983 y 1994 a través de varios proyectos.

La metodología CommonKADS sigue una aproximación al desarrollo de SBC como la construcción de un número de modelos interrelacionados que capturan los principales rasgos del sistema y de su entorno. El proceso de desarrollo de SBC consiste en rellenar un conjunto de “plantillas” de los modelos.

### **2.4.2 LIMITACIONES DE COMMONKADS PARA SISTEMAS MULTIAGENTE**

En primer lugar, debemos hacer notar que CommonKADS no fue diseñado para desarrollar un SMA. Los principales problemas encontrados para aplicar directamente CommonKADS al desarrollo de SMA son [IGL98]:

- ◆ La metodología CommonKADS gira alrededor del modelo de la experiencia y está pensada para desarrollar sistemas expertos que interactúan con un usuario. El resto de modelos, en especial los modelos de agente, comunicación y organización, no son suficientemente desarrollados e integrados en la metodología. Dado que la metodología está concebida para un sistema experto centralizado.
- ◆ La concepción de agente dentro de CommonKADS es muy restrictiva, normalmente habrá dos agentes básicos, el usuario y el sistema experto. El concepto de agente de CommonKADS no exhibe ninguna de las propiedades de los agentes (proactividad, reactividad, sociabilidad, adaptación, etc.), simplemente es empleado para delimitar el sistema experto de los usuarios y sistemas externos.
- ◆ El Modelo de Comunicación trata principalmente de interacciones hombre-máquina. Es muy restrictivo para interacciones entre software. De la misma manera que en el Modelo de Interacción, solo se modelan las entradas y salidas de las tareas llevadas a cabo entre un usuario y un SBC.
- ◆ El Modelo de Comunicación no permite describir las transacciones entre varios agentes de una forma natural.
- ◆ Aunque las relaciones entre los agentes pueden describirse entre los agentes en el Modelo de Organización, dicho modelo estaba previsto para analizar relaciones principalmente humanas.

### 2.4.3. MAS-COMMONKADS

La metodología fue la primera en hacer un planteamiento de SMA integrado con un ciclo de vida de software, concretamente el espiral dirigido por riesgos. Considera sólo dos agentes básicos: el usuario y el sistema. Lo que influye en el modelo de comunicación que, consecuentemente, trata de interacciones hombre-máquina. Propone siete modelos para la definición del sistema: Agente, tareas, experiencia, coordinación, comunicación, organización y diseño. Cada modelo presenta una reseña a la teoría sobre la que se basa. El modelo en sí parte de una descripción gráfica que luego se complementa con explicaciones en lenguaje natural de cada elemento. Existe por cada modelo una descripción de las dependencias respecto de otros modelos y de las actividades involucradas. Los modelos se hayan descritos ampliamente en lenguaje natural, complementándose con otras notaciones como SDL (*Specification and Description Language*) [International Telecommunication Union] o MSC (*Message Sequence Chart*) [International Telecommunication Unión] para describir el comportamiento de los agentes cuando interactúan.

La especificación de SMA que proporciona MAS-CommonKADS detalla la mayoría de aspectos en lenguaje natural. Esta particularidad dificulta el análisis automático de la especificación generada y supone una gran desventaja frente a semi-formalismos como UML, el problema se salva utilizando meta-modelos [IGL98] como mecanismo de especificación. El desarrollo de meta-modelos está soportado por herramientas que permite el procesamiento automático de los modelos generados. Los meta-modelos en algunos casos profundizan más en el detalle que MAS-CommonKADS. Tal es el caso del meta-modelo de organización, el de tareas y objetivos, o el de agentes.

### 2.4.4. CARACTERÍSTICAS DE MAS-COMMONKADS

Documento 1 (Clave:Doc1)	Características de MAS- CommonKADS
a) <b>Ciclo de vida</b>	El modelo de ciclo de vida propuesto para desarrollar los modelos de MAS-CommonKADS es el modelo en espiral dirigido por riesgos es decir dirigido por casos de uso.
b) <b>Perspectiva de Desarrollo</b>	La metodología a lo largo del desarrollo de los diferentes modelos sigue una perspectiva de desarrollo de <b>Top-Down</b> (desarrollan primero las partes generales y después los detalles esas partes), y la perspectiva <b>Bottom-Up</b> (primero se resuelven los sub-problemas que son usados para solucionar los problemas mayores)
c) <b>Dominio de Aplicación</b>	La metodología MAS-CommonKads ayuda desarrollar SMA's en ámbitos como Ciencias sociales y ciencias naturales, también tecnología informática y de telecomunicación. Es decir la metodología es independiente debido a que es aplicable

		a cualquier dominio.
<b>d) Tamaño de SMA</b>	<b>de</b>	Un Sistema Multi-Agente se conforma a partir de dos agentes, así el mínimo es de dos agentes, el máximo para un SMA desarrollo con MAS-CommonKADS, después de revisar bibliografía [HEGI05] y paginas Web el tamaño no es especificado en ninguna de estas dos fuentes.
<b>e) Tipos de Agentes</b>	<b>de</b>	MAS-CommonKADS puede soportar el a Agentes Móviles (capaces de salir de los límites de las redes y acceder a computadores en redes remotas o incluso Internet), Agentes Reactivos, Agentes Deliberativos, y Agentes Híbridos (combinación de agentes reactivos (estímulos) y agentes deliberativos (con un modelo interno de razonamiento)).
<b>f) Soporte para comprobación y validación</b>		El soporte para comprobación y validación del SMA desarrollado con esta metodología solo se vería si se realiza la implementación con alguna herramienta de software.
<b>g) Enfoque de Desarrollo</b>	<b>de</b>	El enfoque de desarrollo de MAS-CommonKADS es el enfoque orientado a la ingeniería del conocimiento, este enfoque se centra en el desarrollo de sistemas basados en conocimiento, como lo son los agentes. La definición del conocimiento de un agente es un proceso de adquisición de conocimiento, lo cual es abordado por este tipo de método.

Tabla 1. Documento 1(Características de MAS-CommonKADS)

#### 2.4.5. FASES DE DESARROLLO DE MAS-COMMONKADS

El modelo de ciclo de vida para el desarrollo de Sistemas Multi-Agente con MAS-CommonKADS sigue las siguientes fases:

Documento 2 (Clave:Doc2)		Fases de Desarrollo de MAS- CommonKADS
<b>Conceptualización</b>		Tarea de extracción o adquisición de conocimiento (elicitación) para obtener una primera descripción del problema y la determinación de los casos de uso que pueden ayudar a entender los requisitos informales y para probar el sistema.
<b>Análisis</b>		Determinación de los requisitos de nuestro sistema partiendo del enunciado del problema. Durante esta fase se desarrollan los siguientes modelos: Organización, tareas, agente, comunicación, coordinación y experiencia.
<b>Diseño</b>		Determinación de cómo los requisitos de la fase de análisis pueden ser logrados mediante el desarrollo del modelo de diseño. Se determinan las arquitecturas tanto de la red Multi-Agente como de cada agente.

Tabla 2. Documento 2(Fases de Desarrollo de MAS-CommonKADS)

#### 2.4.6. MODELOS DE MAS-COMMONKADS

MAS-CommonKADS propone los siguientes modelos para el desarrollo de SMA's

- ◆ Modelo de Agente (AM)
- ◆ Modelo de Organización (OM)
- ◆ Modelo de Tareas (TM)
- ◆ Modelo de la Experiencia (EM)
- ◆ Modelo de Comunicación (CM)
- ◆ Modelo de Coordinación (CoM)
- ◆ Modelo de Diseño (DM)

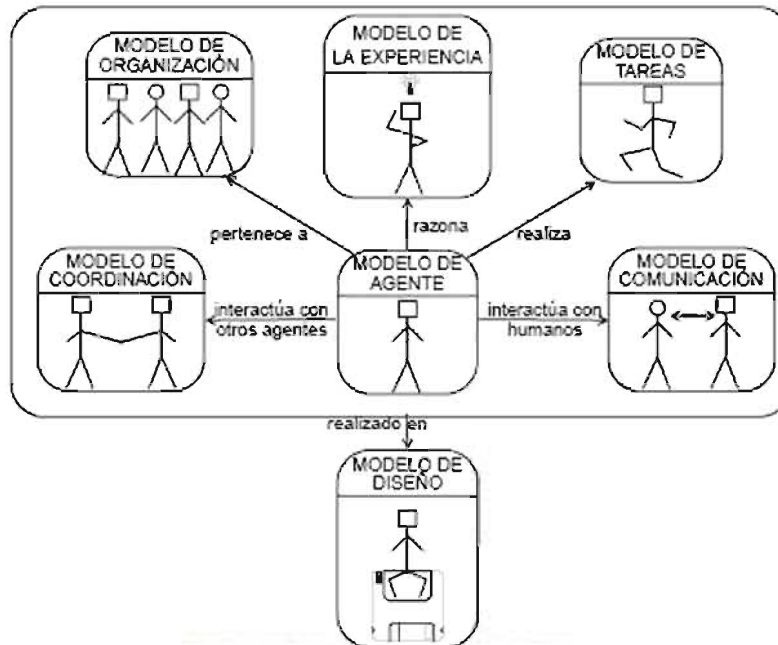


FIGURA 2.5. Los modelos de MAS-CommonKADS

Fuente [IGL98]

Tabla 3. Documento 3 (Modelos de MAS-CommonKADS)

#### 2.4.7. MODELO DE AGENTE

Los agentes son capaces de realizar tareas, que se especifican en un ejemplar del modelo de tareas. Los agentes pueden ofertar la realización de determinadas tareas (denominadas servicios) a otros agentes. Los agentes se caracterizan además por tener objetivos. Un objetivo es una responsabilidad aceptada por el agente. Los objetivos pueden adoptarse desde el nacimiento del agente (objetivos iniciales) o en el transcurso de la ejecución del agente.

Los agentes pueden adoptar objetivos para satisfacer otros objetivos propios, para satisfacer objetivos de otros agentes, o para responder a eventos del entorno. La principal distinción para clasificar a los agentes es entre agentes software y agentes humanos.

Mientras que no se puede esperar la captura completa de las habilidades y restricciones de los agentes humanos, sí deben especificarse para los agentes software.

El modelo de agente tiene como propósito: “La especificación de los diversos agentes involucrados en la resolución del problema” [IGL98].

El modelo de agente sirve para especificar las características de los agentes involucrados en la resolución de un problema. El modelo de agente está pensado para recoger los requisitos que debe tener un agente para poder realizar las tareas (responsabilidades) asignadas. El modelo de agente es el modelo central de MAS-CommonKADS y debe ser incluido siempre.

#### 2.4.7.1. Ciclo De Desarrollo Del Modelo De Agente

El modelo de agente es una herramienta para analizar el sistema desde una perspectiva orientada a agentes. A continuación las características de desarrollo de modelo de agente

- ◆ **Actividad:** Identificación de los agentes
- ◆ **Descripción:** En esta fase se emplean diversas técnicas para identificar los agentes del sistema.

Agente:				Clase:	
Objetivos	Planes	Conocimiento	Colaborador	Servicio	

FIGURA 2.6. Tarjetas CRC para Agentes  
Fuente [IGL98]

- ◆ **Estados resultantes:** Agentes identificados. Utilización de heurísticos para identificar los agentes. Determinación de los agentes utilizando “casos de uso internos”, identificación de agentes empleando tarjetas CRC (Clases-Responsabilidades-Colaboraciones).
- ◆ **Actividad:** Descripción de los agentes, Identificación de los objetivos procedentes del modelo de tareas, identificación de los objetivos empleando tarjetas CRC, identificación de restricciones, identificación de servicios.

#### 2.4.8. MODELO DE TAREAS

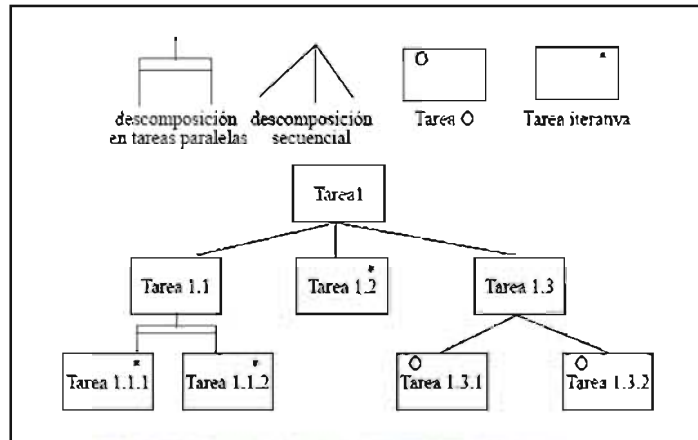
El modelo de tareas describe las tareas que los agentes pueden realizar, los objetivos de cada tarea, su descomposición, los ingredientes y los métodos de resolución de problemas para resolver cada objetivo, permite describir las actividades relacionadas



para alcanzar un objetivo. El objetivo del desarrollo del modelo de tareas es documentar la situación actual y futura de la organización, facilitar la gestión de cambios, y ayudar a estudiar el alcance y viabilidad del sistema inteligente que se desea desarrollar

### 2.4.8.1. Desarrollo y Gestión Del Modelo De Tareas

El modelo de tareas define las situaciones y objetivos de riesgo que permiten determinar qué estados de los constituyentes del modelo de tareas se desean alcanzar.



**FIGURA 2.7.** Notación alternativa para representar la descomposición de tareas  
Fuente [IGL98]

El desarrollo del modelo de tareas no es específico de CommonKADS, sino que es una tarea general para cualquier metodología de ingeniería software.

El modelado de una de las tareas debe terminarse cuando:

- ◆ Puede realizarse de una forma simple y bien determinada
- ◆ Puede ser realizada por software estándar.
- ◆ Requiere conocimiento y se puede refinar mejor en el modelo de la experiencia.
- ◆ Indica comunicación y se puede refinar en el modelo de comunicación o coordinación.
- ◆ Y una tarea no puede descomponerse en tareas con una estructura de control simple. Si es así será candidata de refinarse en el modelo de la experiencia

### 2.4.9. MODELO DE EXPERIENCIA

Se presenta el modelo de la experiencia describe el conocimiento necesitado por los agentes para alcanzar sus objetivos. Sigue la descomposición de *CommonKADS* y reutiliza las bibliotecas de tareas genéricas para desarrollar y describir el conocimiento de los agentes MAS-CommonKADS, desarrolla en el modelo de la experiencia las tareas que

requieren conocimiento para ser llevadas a cabo y que permitirán caracterizar al agente como un sistema basado en conocimiento.

#### 2.4.9.1. Estructura Del Modelo De Experiencia

Los principales componentes del modelo de la experiencia [IGL98] se muestran en la Figura 2.8. Esta estructura recoge los cuatro niveles del conocimiento de KADS-I (dominio, inferencia, tarea y estrategia) [IGL98] al que se unen los métodos de resolución de problemas.

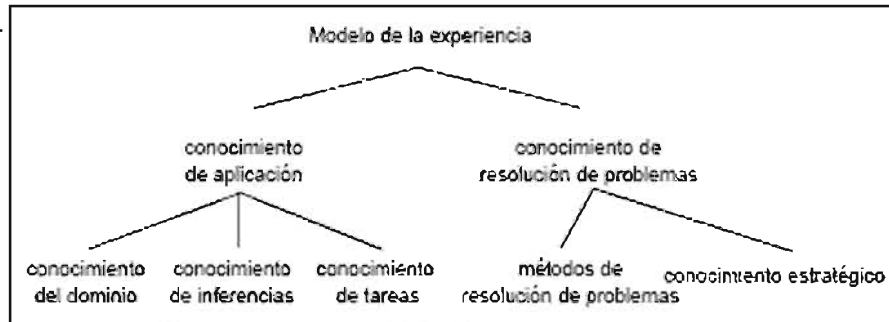


FIGURA 2.8. Principales componentes del modelo de la experiencia  
Fuente [IGL98]

#### 2.4.9.2. Técnicas De Adquisición Del Conocimiento

Las técnicas de adquisición del conocimiento permiten que el ingeniero del conocimiento pueda modelar el conocimiento empleado por humanos en la realización de actividades. Se pueden distinguir tres tipos de técnicas:

- a) **Técnicas de entrevistas:** Técnicas verbales para entrevistar a un experto, al que se le solicita que reflexione sobre su forma de resolver los problemas y la explique. Estas técnicas pueden producir transcripciones.
- b) **Técnicas de observación:** Técnicas basadas en observar cómo resuelve el experto un problema real o simulado. Estas técnicas pueden producir transcripciones.
- c) **Técnicas multidimensionales:** Técnicas que producen datos no verbales, forzando que el experto piense sobre el dominio desde una nueva perspectiva.

#### 2.4.10. MODELO DE COORDINACIÓN

El modelo de coordinación tiene como propósito almacenar las decisiones sobre la estructura de comunicaciones y los protocolos de dichas comunicaciones, abordando los riesgos asociados con los requisitos funcionales de los protocolos de comunicación.

El modelo de coordinación describe las interacciones entre agentes involucrados en la resolución de un problema. El modelo estructura las interacciones en conversaciones, cada interacción entre dos agentes se realiza mediante el envío de un mensaje. El modelo

de coordinación propuesto está orientado a servicios: Un agente puede ofertar la realización de determinadas tareas, denominadas *servicios*, a otros agentes. La oferta y demanda de los servicios se realiza con tipos de mensajes especiales y mediante el mensaje *request-register* según la especificación de la FIPA (sección 2.2.1.) En una conversación se distinguen los siguientes papeles desempeñados por los agentes: el *iniciador*, agente que inicia la conversación y *participantes*, agentes involucrados en la conversación. En MAS-CommonKADS el modelo de comunicación sólo se encarga del modelado de las relaciones hombre-máquina. Por tanto las interacciones con otros agentes no humanos son modeladas en el modelo de coordinación. Las señales recibidas del exterior son modeladas mediante una entidad genérica denominada entorno.

#### 2.4.10.1. Estructura Del Modelo

Los constituyentes del modelo se estructuran en torno a los conceptos de agente y tareas (objetivos) e introduce los siguientes conceptos:

- **Mensaje:** Estructura de datos que intercambian los agentes para comunicarse.
- **Acto de habla:** Intención del emisor del mensaje al transmitir el contenido del mensaje.
- **Servicio:** Prestación realizada por un agente para satisfacer las necesidades de otro agente.
- **Intervención:** Intercambio de un mensaje entre un agente y otro agente. Puede haber tres tipos de interacciones básicas: síncrona (bloqueante hasta recibir respuesta), asíncrona (sin respuesta) y diferida (envío no bloqueante y recepción de respuesta).
- **Conversación:** Conjunto de interacciones cuyo fin es la realización de un servicio. Distinguimos dos tipos de conversaciones:
  - ◆ conversaciones de información.
  - ◆ conversaciones de servicio.

Las conversaciones de información tienen como objetivo informar a un agente de algún hecho u obtener una determinada información (consulta). Las conversaciones de servicio tienen como objetivo la realización de un servicio.

- **Protocolo:** Normas asociadas a las conversaciones. Un protocolo determina las interacciones que se llevan a cabo en una conversación.

#### 2.4.11. MODELO DE ORGANIZACION

El modelo de organización de MAS-CommonKADS tiene como objetivo analizar desde una perspectiva de grupo las relaciones entre los agentes (tanto software como

humanos) que interactúan con el sistema, es decir es una herramienta para analizar la organización humana en que el Sistema Multi-Agente será introducido y para describir la organización de los agentes software y su relación con el entorno.

#### **2.4.11.1. Ciclo De Desarrollo Del Modelo**

El desarrollo estándar del modelo de organización está dividido en dos estados: el estudio de la organización humana y el estudio de la organización Multi-Agente. En el caso de que se realicen ambos, deben abordarse en el siguiente orden:

- a) Modelado de la organización humana. Los principales objetivos del modelado de la organización humana son la comprensión de su funcionamiento, la detección de los problemas que pueden ser resolubles con un sistema inteligente, la propuesta de soluciones y el estudio de la viabilidad de dichas propuestas.
- b) Modelado de la sociedad Multi-Agente. Los principales objetivos del modelado de una sociedad Multi-Agente son la descripción de las relaciones entre los agentes y con su entorno.

#### **2.4.11.2. Actividades De Construcción Del Modelo De Organización**

Se puede distinguir dos tipos de actividades del modelo de organización:

- a) Actividades relacionadas con el modelado de la organización en que se introduce el sistema, orientadas al estudio de la viabilidad del desarrollo del sistema y del impacto del sistema en la organización.
- b) Actividades relacionadas con el modelado de la estructura organizativa de la sociedad de agentes, orientadas a la descripción de dicha sociedad.

#### **2.4.12. MODELO DE DISEÑO**

Frente a la concepción tradicional de que el diseño transforma un conjunto de requisitos, que sólo especifican qué debe hacer el sistema, en la descripción de componentes agrupados en módulos; mientras que los otros cinco modelos tratan del análisis del Sistema Multi-Agente, el modelo se utiliza para describir la arquitectura y el diseño del Sistema Multi-Agente como paso previo a su implementación

No contienen sólo los requisitos funcionales, sino que también contienen requisitos no funcionales (tales como el tiempo permitido de ejecución de una tarea) y toman decisiones que podrían ser consideradas de diseño (por ejemplo, el modelado de un razonamiento como causal o heurístico en el modelo de la experiencia).

Las ventajas que se pretenden obtener a través del desarrollo del modelo de diseño son:

- ◆ Separación entre análisis e implementación.

- ◆ Especificación del sistema independiente de la plataforma.
- ◆ Descomposición de la tarea de diseño.
- ◆ Adición de requisitos del entorno.
- ◆ Reutilización de componentes.
- ◆ Diseño de la interfaz y la interacción de grupos de agentes

#### 2.4.12.1. Estructura Del Modelo

El modelo de diseño de MAS-CommonKADS distingue tres clases de decisiones de diseño (Figura 2.9.): Diseño de la red, que consiste en diseñar el modelo de red, diseño de los agentes, que consiste en descomponer cada agente en subsistemas, y diseño de la plataforma, que recoge las decisiones de software y hardware.



FIGURA 2.9. Descomposición del modelo de diseño  
Fuente [IGL98]

#### ◆ Diseño del modelo de red

Para determinar las funcionalidades del modelo de red y los agentes de red encargados de llevarlas a cabo, deberemos tomar las siguientes decisiones para cada nivel del modelo:

- a) **Nivel de Red**, se toman las decisiones de diseño sobre la infraestructura de la arquitectura multi-agente. Se deben especificar qué agentes se necesitan para mantener la sociedad multi-agente.

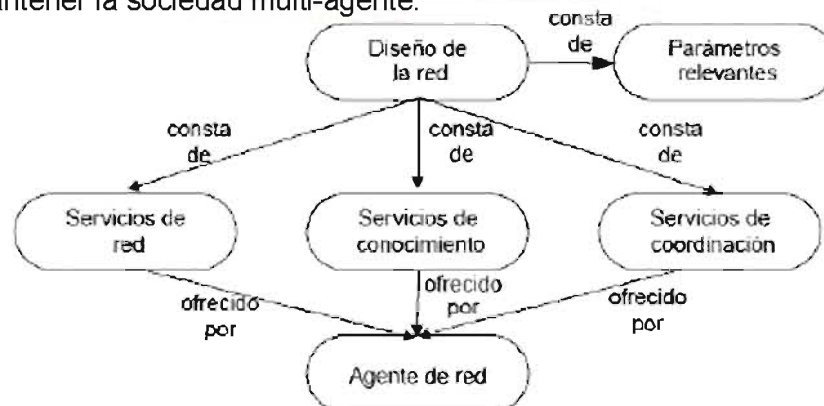


FIGURA 2.10. Constituyentes del diseño de red  
Fuente [IGL98]

- b) **Nivel de Conocimiento**, varias decisiones de diseño: ¿Deben los agentes entender diferentes lenguajes de representación del conocimiento? ¿Es necesario un agente traductor entre lenguajes de representación del conocimiento?

c) **Nivel de Coordinación**, el conjunto de primitivas adecuadas y los agentes para formar planes conjuntos son la base de la definición de este nivel.

◆ **Diseño de los agentes**

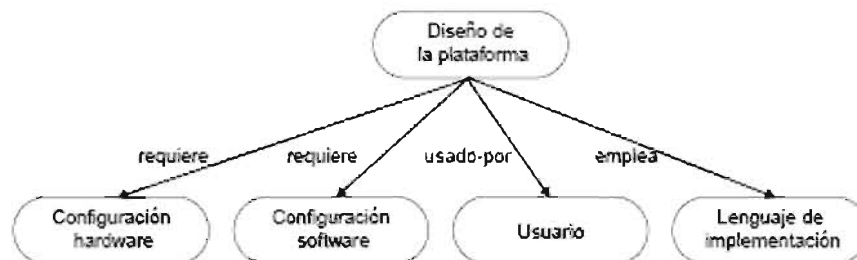
El diseño de los agentes documenta la arquitectura seleccionada para cada agente y las funciones de los módulos que deben implementarse en dicha arquitectura.

Los principales componentes del diseño de los agentes son:

◆ **Diseño de la plataforma**

El diseño de la plataforma permite documentar las decisiones de bajo nivel sobre el lenguaje de implementación seleccionado, el software y hardware empleado y los usuarios finales del sistema.

Los principales componentes del diseño de la plataforma son:



**FIGURA 2.11.** Diseño de la plataforma  
Fuente [IGL98]

### 2.4.13. MODELO DE COMUNICACIÓN

MAS-CommonKADS modelo de comunicación que describa los actos de habla involucrados en las conversaciones desarrolladas en el modelo de coordinación. Así describe todas las comunicaciones directas a través del intercambio de información usando mecanismos del tipo pase de mensajes, y comunicaciones indirectas a través del depósito de información en objetos (uso de estrategias de tipo memoria compartida y métodos de estímulo-respuesta), que reflejan las interacciones entre los agentes.

Los atributos del modelo de comunicación permiten especificar por cada acto de habla involucrado en una conversación, lo siguiente:

- ◆ nombre (Nombre del acto de habla)
- ◆ objetivo (objetivo del acto de habla)
- ◆ Tipo (asertivo, directivo, compromisivo, expresivo o declarativo)
- ◆ Agentes Participantes (agentes que participan en el acto de habla)
- ◆ Comunicación (un acto de habla puede ser realizado de manera directa o indirecta.)
- ◆ Emisor (emisor del mensaje), Receptor (receptor del mensaje)
- ◆ Conversación (nombre de la conversación donde participa)

- ◆ Servicio (nombre del servicio asociado),
- ◆ Datos Intercambiados (descripción de los datos intercambiados en el acto de habla)
- ◆ Descripción (descripción del acto de habla en lenguaje natural)
- ◆ Precondición (condiciones que deben darse en el emisor y receptor para que el acto de habla pueda darse con éxito.)
- ◆ Condición de Terminación (condiciones en el emisor y receptor tras el desarrollo del acto de habla)
- ◆ Per formativa (verbo que describe la intención del acto de habla),
- ◆ Per formativas de intervención
- ◆ Per formativas de red
- ◆ Medio de Comunicación (vía de comunicación).

## 2.5. METODOLOGIA INGENIAS

INGENIAS es la evolución de las ideas de MESSAGE<sup>2</sup>. INGENIAS se basa en la integración de los resultados de investigación en el ámbito de las tecnologías de desarrollo de agentes con el RUP<sup>3</sup>, que es un conocido proceso de desarrollo de software (usualmente aplicado a desarrollos Orientados a Objetos).

INGENIAS mejora MESSAGE en tres aspectos:

- ◆ Integración de las vistas de diseño del sistema
- ◆ Integración de resultados de investigación
- ◆ Integración con el ciclo de vida de desarrollo de software

La metodología propone un lenguaje de especificación de Sistemas Multi-Agente siguiendo el ejemplo de MESSAGE, el lenguaje se especifica con meta-modelos que indican qué hace falta para describir: agentes aislados, organizaciones de agentes, el entorno, interacciones entre agentes o roles, tareas y objetivos. Los meta-modelos mencionados antes se construyen mediante un lenguaje natural de meta-modelado (GOPRR<sup>4</sup>). La ejecución de actividades para producir modelos se basa en la herramienta INGENIAS IDE, una herramienta para modelado visual. La herramienta almacena la especificación del sistema utilizando XML. Desde la especificación en XML, se plantea el

---

2. Methodology for Engineering Systems of Software AGENTS, proyecto que extiende ingeniería de software OO con conceptos del área de agentes.

3. Rational Unified Process, el RUP es un proceso de ingeniería de software, que utiliza el paradigma de orientación a objetos para su descripción, en un marco de proceso configurable para satisfacer necesidades específicas, e implementa las mejores prácticas de desarrollo de software.

4. Graph, Object, Property, Relationship, and Role (Gráficos, Objetos, Propiedades, Relaciones y Roles)

procesarla para generar código o para generar la documentación.

### 2.5.1. CARACTERÍSTICAS DE INGENIAS

Documento 4 (Clave:Doc4)		Características de INGENIAS
a) Ciclo de vida		El modelo de ciclo de vida propuesto para desarrollar los modelos de INGENIAS es el Proceso Unificado de Desarrollo que esta basado en componente que está dirigido por casos de uso, centrado en la arquitectura, y es iterativo e incremental.
b) Perspectiva de Desarrollo		Los diferentes modelos de Ingenias sigue una perspectiva de desarrollo de <b>Top-Down</b> y la perspectiva <b>Bottom-Up</b> , combinando las dos perspectivas se tiene una Híbrida. (Ver Sección 2.5.8 de este capítulo Modelo de Interacción)
c) Dominio de Aplicación		INGENIAS ayuda a desarrollar SMA's en ámbitos como el análisis experimental, teórico de mecanismos de autoorganización y emergencia que surgen cuando varias entidades autónomas interaccionan. También en creación de artefactos distribuidos capaces de realizar tareas complejas mediante cooperación e interacción. INGENIAS es independiente en cuanto al dominio de aplicación debido a que es aplicable a cualquier dominio pero si fuera aplicable a un dominio en específico sería de dependiente.
d) Tamaño de SMA		Según [HEGI05] y en el sitio oficial de INGENIAS ( <a href="http://grasia.fdi.ucm.es/Ingenias">http://grasia.fdi.ucm.es/Ingenias</a> ) el tamaño no es especificado.
e) Tipos de Agentes		INGENIAS puede soportar Agentes Reactivos, Agentes Deliberativos, y Agentes Híbridos.
f) Soporte para comprobación y validación		El soporte para comprobación y validación del SMA desarrollado con esta metodología es vista y plasmada con su herramienta de desarrollo INGENIAS IDK.
g) Enfoque de Desarrollo		INGENIAS es una metodología que ha evolucionado de la tecnología orientadas a objetos. Los agentes y los objetos tienen mucho en común, incluso han sido considerados como objetos activos. Algunas de las técnicas para identificar objetos también pueden ser usadas para identificar agentes.

Tabla 4. Documento 4 (Características de INGENIAS)

### 2.5.2. NOMENCLATURA

El nombre asociado a las relaciones obedece a unas reglas sencillas mostradas. Se trata de hacer que el nombre de la relación sea precedido por un conjunto de letras que denote su procedencia, como el flujo de trabajo (WF), meta-modelo de agente (A), interacción (I), unidad de interacción (UI), modelos de tareas y objetivos (GT), relaciones sociales (AGO), organización (O) o el entorno (E).



### 2.5.3. ENTIDADES BÁSICAS

La metodología proporciona una jerarquía de conceptos básicos para el desarrollo de un SMA así como una notación para representar los conceptos. La jerarquía comienza con la Entidad MAS GRASIA y la Relación MAS GRASIA, que reciben ese nombre por el grupo de investigación en que han sido desarrolladas (**GR**upo de **Ag**entes Software del departamento de **S**istemas Informáticos y **proGR**Amación o GRASIA)

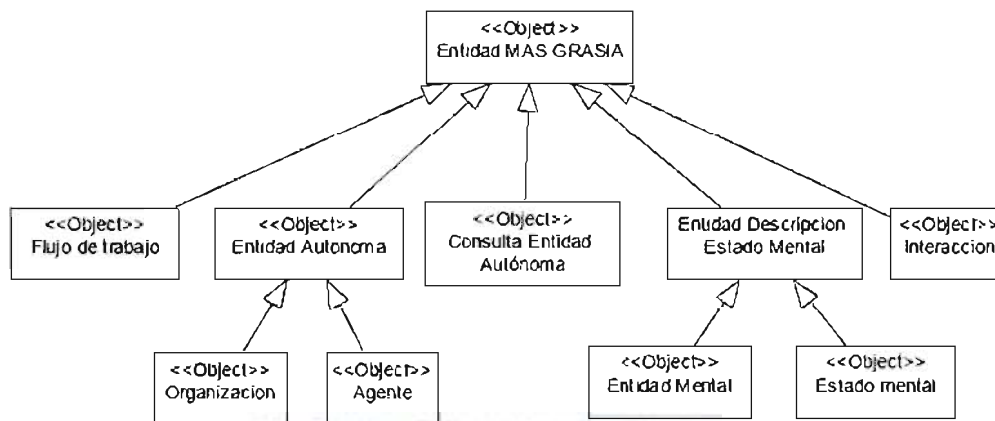


FIGURA 2.12. Entidades básicas de la metodología  
Fuente [GRA02]

### 2.5.4. NOTACIÓN

Los diagramas correspondientes al nivel del modelo posteriormente son exportados a la herramienta de trabajo INGENIAS IDE, los diagramas es la notación:

Documento 5 (Clave:Doc5)	Componentes Notaciones de INGENIAS
	<b>Objetivo.</b> Se etiqueta con el nombre del objetivo
	<b>Rol.</b> Se etiqueta con el nombre del rol
	<b>Procesador de estado mental.</b> Se etiqueta con el nombre del procesador
	<b>Gestor de Estado Mental.</b> Se etiqueta con el nombre del gestor.
	<b>Agente.</b> Se etiqueta con el nombre del agente.
	<b>Grupo.</b> Se etiqueta con el nombre del grupo.
	<b>Organización.</b> Se etiqueta con el nombre del organización.
	<b>Flujo de Trabajo.</b> Se etiqueta con el nombre del flujo.

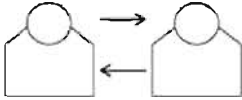
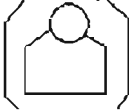
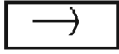

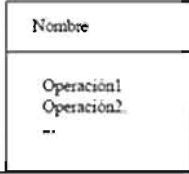
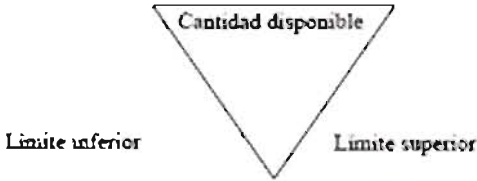

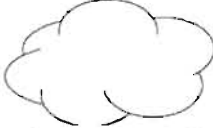
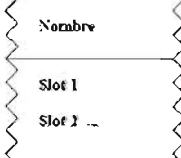
	<p><b>Interacción.</b> Se etiqueta con el nombre de la interacción y su naturaleza como coordinación, planificación o negociación.</p>
	<p><b>Consulta de entidades autónomas.</b> Se etiqueta con el nombre de agentes existentes o expresiones que denotan agentes existentes.</p>
	<p><b>Unidad de Interacción.</b> Se etiqueta con el nombre de la unidad de interacción.</p>
	<p><b>Tarea.</b> Se etiqueta con el nombre de la tarea.</p>
	<p><b>Aplicación.</b> Se etiqueta con el nombre de la aplicación y las operaciones soportadas.</p>
	<p><b>Recurso.</b> Se etiqueta con el nombre del recurso, la cantidad disponible del mismo, el límite inferior y superior admisible. Por debajo o encima de estos límites, el recurso y deshabilitado</p>
	<p><b>Hecho.</b> Se etiqueta con el nombre de hecho y los nombres de los slots identificados.</p>
	<p><b>Creencia.</b> Se etiqueta con el nombre de la evidencia e información acerca de que es lo que se esta aceptando como cierto.</p>
	<p><b>Evento.</b> Se etiqueta con el nombre del evento y los nombres de los spot identificados.</p>

Tabla 5. Documento 5 (Componentes Notacionales de INGENIAS)

### 2.5.5. EL LENGUAJE DE ESPECIFICACIÓN

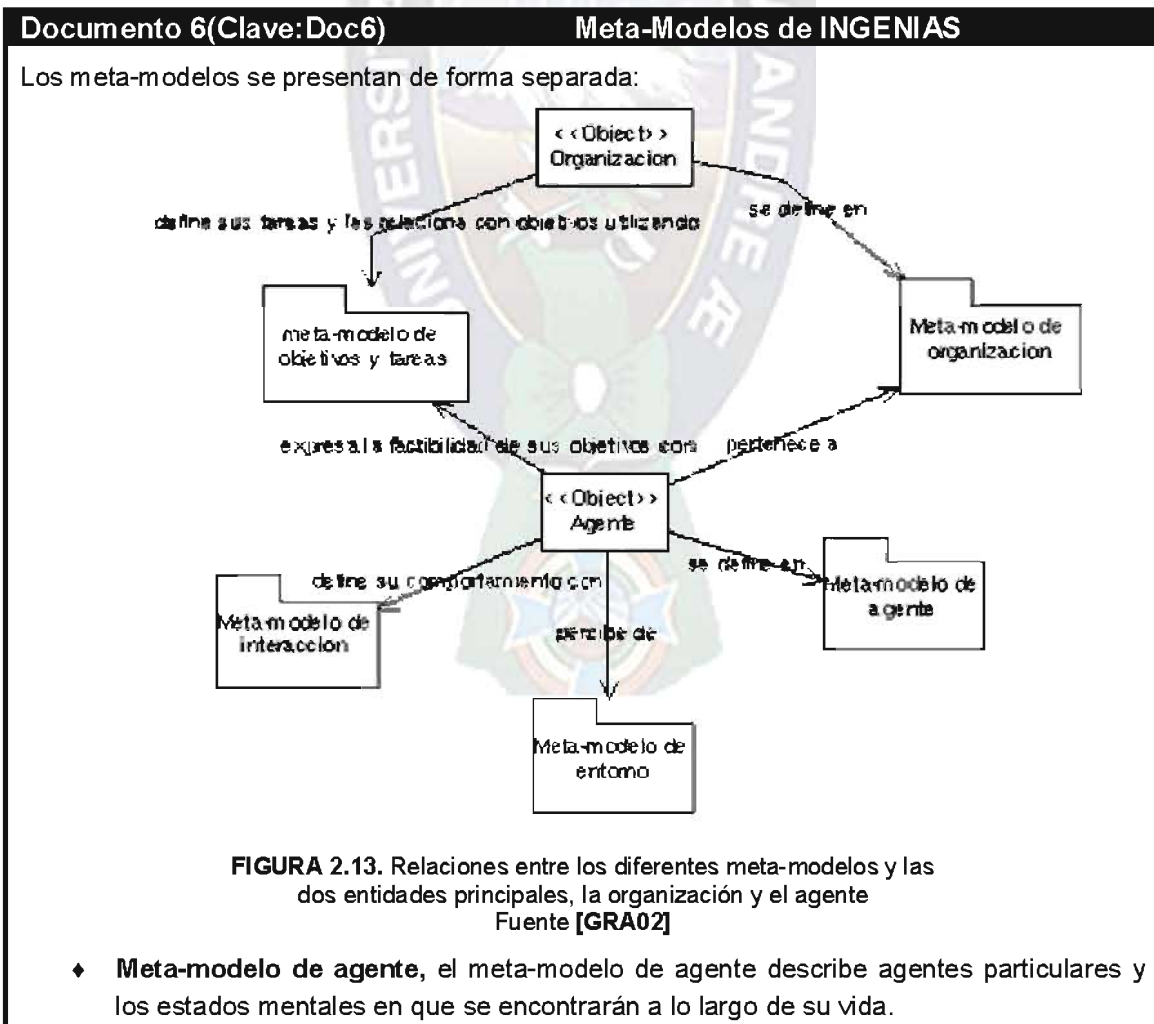
El método de desarrollo de SMA propuesto en INGENIAS concibe el SMA como la representación computacional de un conjunto de modelos. Cada uno de los modelos muestra una visión parcial del SMA: los agentes que lo componen, las interacciones que

existen entre ellos, cómo se organizan para proporcionar la funcionalidad del sistema, qué información es relevante en el dominio y cómo es el entorno en el que se ubica el sistema a desarrollar.

### 2.5.5.1. Meta-Modelos

Un *meta-modelo* define las primitivas y las propiedades sintácticas y semánticas de un *modelo*. Los *modelos* crecen incorporando más detalle gracias a que no es necesario que se instancien absolutamente todos los elementos del *meta-modelo* para tener un modelo. Este tipo de notación facilita enormemente el desarrollo de sistemas. Otra ventaja de utilizar *meta-modelos* es que las especificaciones generadas de SMA son lo suficientemente estructuradas para ser procesadas de forma automática, es decir que un meta-modelo es una representación de los tipos de entidades que pueden existir en un modelo, sus relaciones y restricciones de aplicación.

Para especificar cómo tienen que ser los modelos en INGENIAS se definen cinco meta-modelos que giran alrededor de dos entidades la *organización* y el *agente*.



- ♦ **Meta-modelo de tareas y objetivos**, se usa para asociar el estado mental del agente con las tareas que ejecuta.
- ♦ **Meta-modelo de organización**, define cómo se agrupan los agentes, la funcionalidad del sistema y qué restricciones hay que imponer sobre el comportamiento de los agentes.
- ♦ **Meta-modelo de interacción**, detallar cómo se coordinan y comunican los agentes.
- ♦ **Meta-modelo de entorno**, define qué existe alrededor del nuevo sistema y cómo lo percibe cada agente.

Tabla 6. Documento 6 (Meta-Modelos de INGENIAS)

### 2.5.6. MODELO DE AGENTE

El meta-modelo de agente se usa para describir agentes particulares excluyendo las interacciones con otros agentes. El meta-modelo se centra en la funcionalidad del agente y en el diseño de su control. En este sentido, proporciona información acerca de los siguientes aspectos:

#### a) Descripción De Agentes Particulares

- ♦ *Funcionalidad del agente*: Responsabilidades (¿Qué tareas sabe ejecutar?, ¿Qué objetivos se compromete a alcanzar?)
- ♦ *Comportamiento*: Control del agente



FIGURA 2.14. Diagrama de descripción de Agente  
Fuente [PAV04]

#### b) Estado Mental

- ♦ *Agregación de entidades mentales*: objetivos, creencias, compromisos, hechos.

#### c) Gestión De Estado Mental

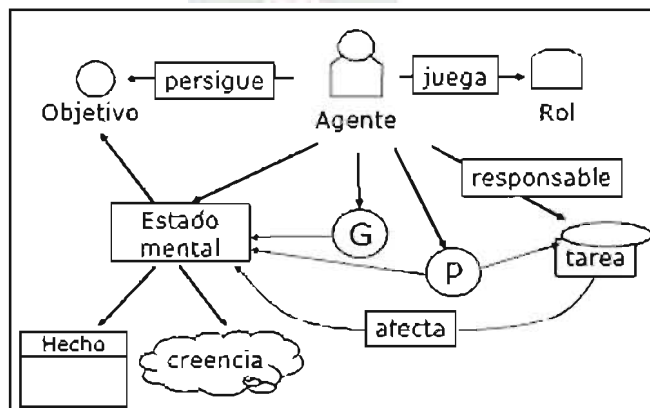
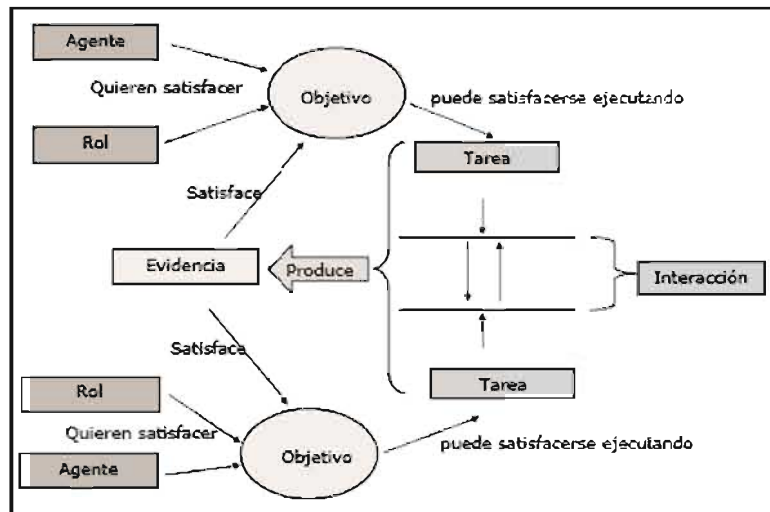


FIGURA 2.15. Diagrama del modelo de agente  
Fuente [PAV04]

- ◆ Creación, destrucción, modificación de las entidades del estado mental.
- ◆ Mecanismo de decisión, procesador de estado mental.
- ◆ Reglas, planificación, etc.



**FIGURA 2.16.** Elementos del modelo de agente  
Fuente [PAV04]

La generación de instancias del meta-modelo se organiza en torno a los resultados esperables durante el análisis y el diseño.

d) Los **resultados** que se obtienen durante el **análisis** son los siguientes:

- ◆ **Funcionalidad del agente.** Un conjunto de roles que desempeña cada agente.
- ◆ **Requisitos del agente.** Que cualidades se esperan del agente

e) Los **resultados** a obtener durante el **diseño** se refieren al control del agente:

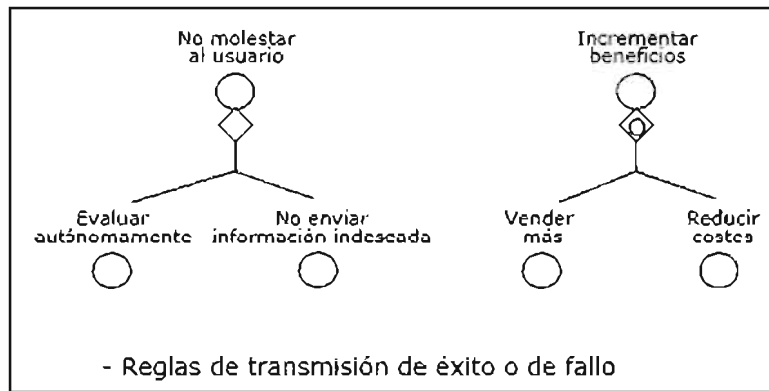
- ◆ **Restricciones de control.** Se representan mediante los estados mentales por los que pasa el agente a lo largo de las interacciones.
- ◆ **Medios de control.** Cómo es el control que asegura la transición entre los distintos estado mentales identificados.

### 2.5.7. MODELO DE OBJETIVOS Y TAREAS

El meta-modelo de interacción refleja el comportamiento deseado de un agente cuando recibe instrucciones de otro agente o bien requiere de otro agente para realizar una tarea. Como en los modelos de agentes, la generación de modelos de instancias se inicia en el diseño.

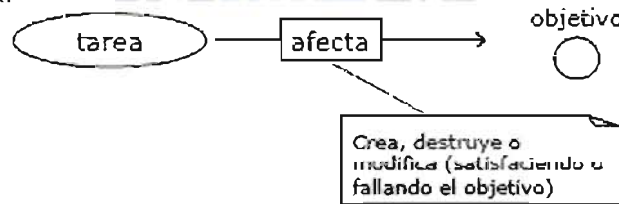
Para el **análisis** se ha fijado que los resultados esperables son:

a) **Tareas y objetivos.** Los objetivos que persigue el sistema, incluidos en los modelos de agente y organización. Se admite la descomposición estructural de tareas y objetivos.



**FIGURA 2.17.** Descomposición de objetivos-Árboles Y/O  
Fuente [PAV04]

b) **Tareas asociadas a objetivos.** La asociación constituye la justificación de la ejecución de la tarea.



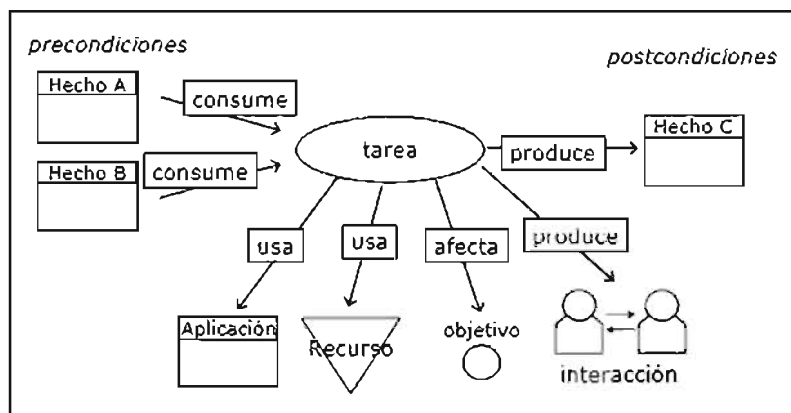
**FIGURA 2.18.** Tareas asociadas a Objetivos  
Fuente [PAV04]

c) **Pre-condiciones y post-condiciones tentativas.** Las precondiciones mínimas son determinar qué entidades mentales se consumen, qué interacciones y entidades mentales se producen y qué aplicaciones se usarán en el proceso.

Para el diseño se ha fijado que los resultados esperables son:

a) **Refinamiento de las dependencias entre objetivos.** Las dependencias pueden desaparecer con motivo de cambios en las dependencias estructurales entre objetivos.

b) **Condiciones de satisfacción o fallo de los objetivos.** Las condiciones se expresan con *patrones de estado mental*.



**FIGURA 2.19.** Elementos de definición de tareas  
Fuente [PAV04]

c) **Pre-condiciones detalladas.** El detalle en las precondiciones viene de añadir los recursos que necesita la tarea y de incluir nuevas instancias.

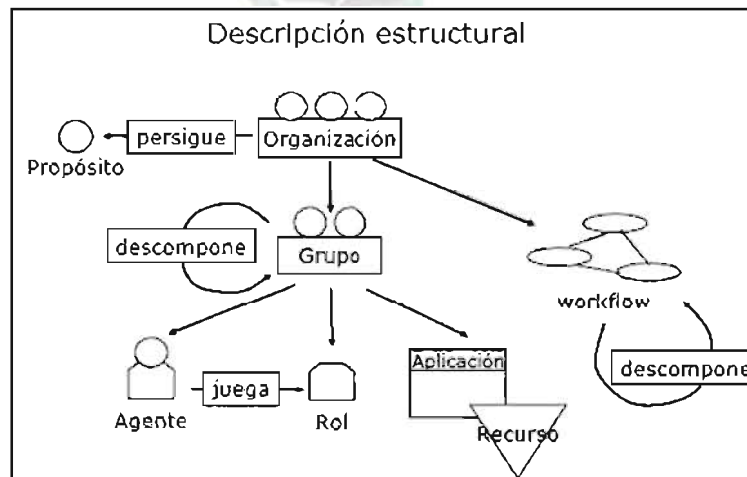
d) **Post-condiciones detalladas.** Se asocian nuevas instancias con recursos para expresar la restitución de los mismos.

### 2.5.8. MODELO DE ORGANIZACIÓN

El meta-modelo de organización es comparable a la arquitectura del sistema en un SMA. El meta-modelo relaciona las entidades activas (agentes, roles), pasivas (aplicaciones, recursos) y los elementos que proporcionan la funcionalidad del sistema (flujos de trabajo, tareas). Por ello, es aconsejable que el meta-modelo sea el primero en estudiarse para generar meta-modelos y que se vuelva a él con frecuencia a medida que van apareciendo nuevos elementos en los demás modelos.

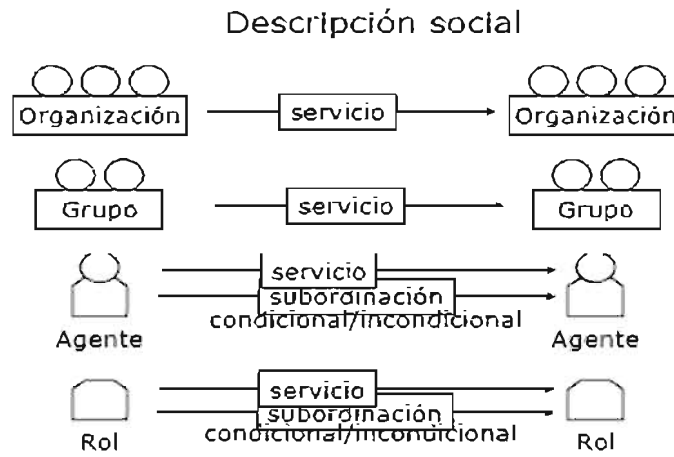
La organización define una **Estructura** donde van a existir los agentes, recursos, tareas y objetivos. La descomposición de la organización es:

- ◆ Grupos
- ◆ Flujos de trabajo
- ◆ Interrelación de tareas en flujos de trabajo
- ◆ Relaciones entre agentes respecto a las tareas
- ◆ Recursos disponibles y asignación
- ◆ Relaciones sociales
- ◆ Relaciones de poder (ej. subordinación) y cliente/servidor entre agentes
- ◆ Relaciones entre grupos
- ◆ Relaciones entre organizaciones
- ◆ Funcionalidad
- ◆ Propósito
- ◆ Tareas que debe realizar



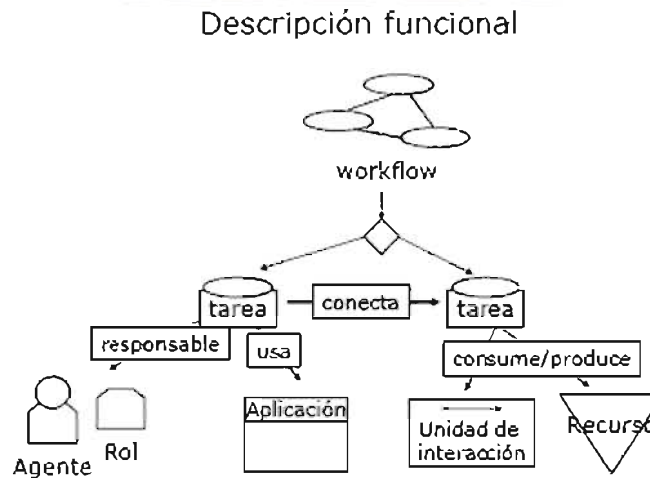
**FIGURA 2.20.** Descripción estructural  
Fuente [PAV04]

La **Descripción Social** en concreto incluye las relaciones de subordinación, de prestación de un servicio y de cliente de un servicio. Las mismas relaciones se definen teniendo como extremos posibles organizaciones, agentes, roles y grupos.



**FIGURA 2.21.** Descripción social  
Fuente [PAV04]

La **Descripción Funcional** describe el objetivo del flujo de trabajo es establecer cómo se asignan los recursos, qué pasos (tareas) son necesarios para la consecución de un objetivo, y quiénes son los responsables de ejecutarlas. En el flujo de trabajo, se habla de *actividades* en lugar de *tareas*, aunque en el contexto actual se puede emplear indistintamente.



**FIGURA 2.22.** Descripción funcional  
Fuente [PAV04]

### 2.5.9. MODELO DE INTERACCIONES

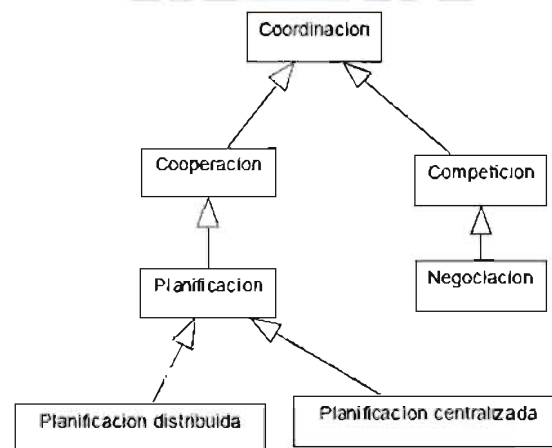
El meta-modelo de interacción refleja el comportamiento deseado de un agente cuando recibe instrucciones de otro agente o bien requiere de otro agente para realizar una tarea. Como en los modelos de agentes, la generación de modelos de instancias se



inicia en el análisis y se continúa en el diseño. El intercambio de conocimiento o peticiones (intencionalidad) entre agentes. Define las interacciones entre los agentes o entre agentes y humanos, se definen a alto nivel, en diseño se detalla el protocolo de interacción.

El Modelo de Interacciones busca simplificar:

- ◆ La definición del contexto de la interacción
- ◆ La generación de una especificación de comportamiento del sistema botton-up / top-down
- ◆ La definición de la ejecución de la interacción.
- ◆ La representación de la interacción.



**FIGURA 2.23.** Naturaleza de la interacción  
Fuente [PAV04]

El modelo de Interacciones se construye sobre: Agentes, roles, objetivos, interacciones y unidades de interacción.

- ◆ Los agentes y los roles son los actores de las interacciones.
- ◆ En las interacciones se ejecutan unidades de interacción (paso de mensajes, lectura y escritura en un espacio de tuplas) en las que hay un iniciador (emisor) y colaboradores (receptores).
- ◆ Se justifica la participación de los actores en la interacción y la existencia de la interacción en sí mediante objetivos.

### **2.5.10. MODELO DE ENTORNO**

Define las entidades del entorno del SMA con las que vaya a interactuar. La mayoría de la información para la generación de instancias del meta-modelo de entorno proviene de la captura de requisitos, que es la entrada para el análisis.

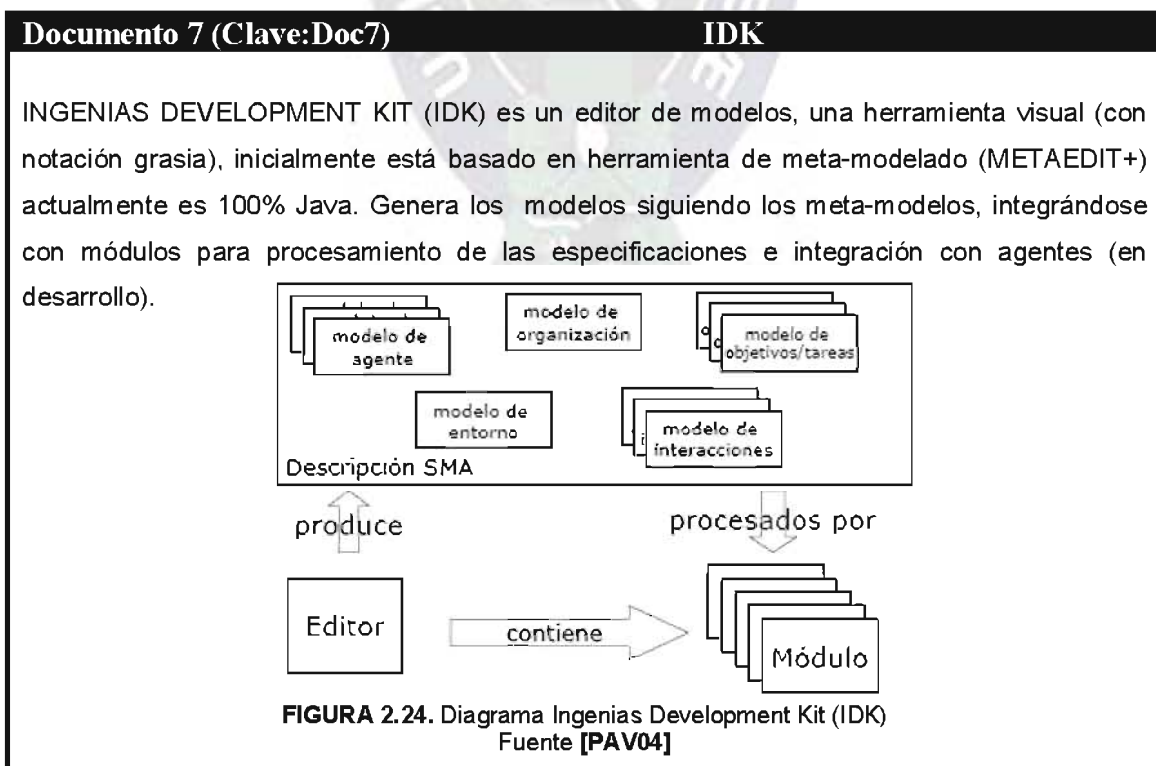
Para el **análisis** se ha fijado que los resultados esperables son:

- ◆ **Entidades del entorno.** Una enumeración de las entidades que preceden al sistema a desarrollar. Estas entidades se categorizan como aplicaciones, agentes o recursos. Cada entidad se acompaña de una descripción de sus funciones, que se entiende han de extraerse de la captura de requisitos.
- ◆ **Percepción del agente.** Se asocia con las aplicaciones del entorno al agente.

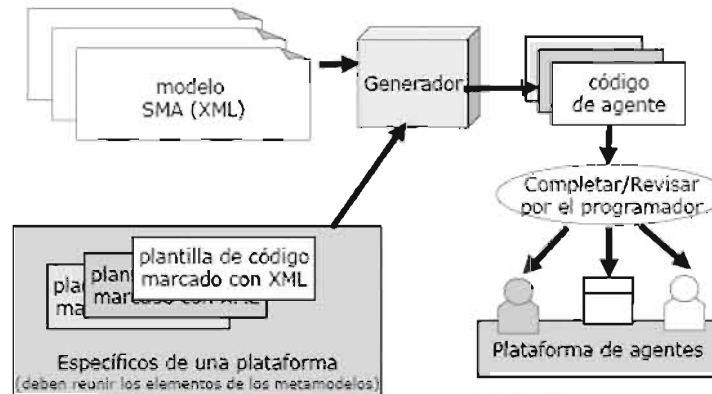
Para el **diseño** se ha fijado que los resultados esperables son:

- ◆ **Configuración de la percepción del agente.** Cada agente asociado con aplicaciones explicita cómo va a percibir, si va a utilizar mecanismos de muestreo o notificación y qué es lo que espera recibir.
- ◆ **Relaciones con tareas.** Aunque se considera en otros meta-modelos, se identifican asociaciones de producción y utilización de aplicaciones y recursos con tareas.
- ◆ **Definición detallada de los recursos.** Consiste en especificar exactamente cómo es cada recurso, detallando su categoría concreta, el límite inferior de consumo, el superior y el estado en que se encuentra inicialmente.

### 2.5.11. INGENIAS DEVELOPMENT KIT (IDK)



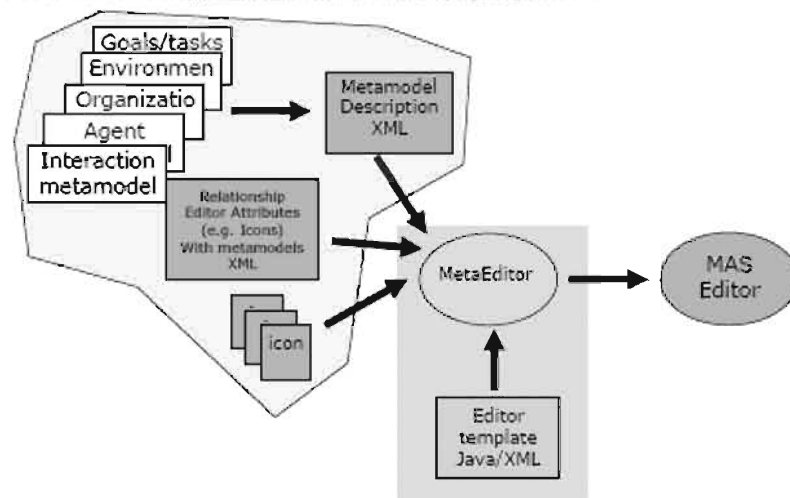
Los módulos son para la generación de código, es decir Armazones (plantillas) configurables, especificados con XML, para distintas plataformas de agentes, como Jade, Robocode, Servlets, Agentes gracia!, que son utilizadas para validar especificaciones: basado en AT y también para generar documentación (HTML) que son el armazón para desarrollar módulos personalizados



**FIGURA 2.25.** Generación de código de IDK  
Fuente [PAV04]

**EDITOR DEL IDK** permite:

- ◆ Crear y modificar modelos de SMA
- ◆ Generar documentación (HTML)
- ◆ Procesar las especificaciones mientras se están generando con el editor o una vez grabadas en un fichero
- ◆ Introducir explicaciones en lenguaje natural de los diferentes diagramas y de cada elemento en los diagramas, así como añadir etiquetas de texto



**FIGURA 2.26.** Generación del editor  
Fuente [PAV04]

**Tabla 7. Documento 7 (IDK)**

## 2.6. JADE

JADE es un entorno de trabajo que simplifica la construcción de Sistemas Multi-Agente que siguen el estándar FIPA (sección 2.2.1. de este capítulo), así como un conjunto de herramientas para la depuración de las plataformas. JADE está implementado en Java para el desarrollo de aplicaciones distribuidas Multi-Agente. JADE proporciona tanto el entorno de desarrollo para la creación de aplicaciones basadas en agentes como el entorno de ejecución.

JADE presenta las siguientes características:

- ◆ P2P: Arquitectura peer-to-peer, cada agente puede tomar la iniciativa en una comunicación o bien responder a peticiones que le hagan otros agentes.
- ◆ Interoperabilidad: JADE cumple con las especificaciones FIPA, por lo que los agentes desarrollados en JADE pueden interactuar con otros agentes que no tienen porque estar desarrollados con JADE, aunque si deben seguir las especificaciones FIPA.
- ◆ Portabilidad La API que proporciona JADE es independientemente de la red sobre la que va a operar, así como de la versión de Java utilizada, teniendo la misma API para J2EE, J2SE y J2ME.
- ◆ Intuitiva: JADE se ha desarrollado para ofrecer una API fácil de aprender y sencilla de manejar.

Los agentes JADE tienen nombres únicos y se permite a cada agente descubrir a otros agentes y comunicarse con ellos mediante comunicaciones punto a punto. Los agentes proporcionan servicios, cada agente puede buscar a otros dependiendo de los servicios que proporcionen otros agentes.

La comunicación entre agentes se lleva a cabo a través de mensajes asíncronos, es decir, el agente que envía el mensaje y el destinatario del mensaje no tienen porqué estar disponibles al mismo tiempo. Es más, el destinatario no tiene porqué existir en ese instante. Los mensajes se pueden enviar a un agente en concreto o se pueden enviar a agentes que se desconocen pero se sabe que poseen unas ciertas características. JADE proporciona mecanismos de seguridad, ya que habrá agentes a los que no se les esté permitido comunicarse con otros agentes, de manera que una aplicación puede verificar la identidad del receptor y del agente que envía el mensaje y no dejar realizar actuaciones no permitidas para un determinado agente. La estructura de los mensajes se basa en el lenguaje ACL (Agent Communication Lenguaje) que ha sido definido por la FIPA.

### 2.6.1. ARQUITECTURA

Los agentes JADE necesitan del entorno de ejecución donde poder "vivir". Cada instancia del entorno de ejecución se denomina *contenedor* (container). Al conjunto de los contenedores se le denomina *plataforma* (platform) y proporciona una capa que oculta a los agentes (y al desarrollador) el entorno donde se ha decidido ejecutar la aplicación.

En cada plataforma debe existir un contenedor especial denominado *contenedor principal* (main container). La principal diferencia del contenedor principal respecto al resto de contenedores es que alberga dos agentes especiales:

- ◆ AMS (Agent Management System), el agente AMS proporciona el servicio de nombres asegurando que cada agente en la plataforma disponga de un nombre único. También representa la autoridad, es posible crear y matar agentes en contenedores remotos requiriendoselo al agente AMS.
- ◆ DF (Directory Facilitator), proporciona el servicio de *Páginas Amarillas*. Gracias al agente DF, un agente puede encontrar otros agentes que provean los servicios necesarios para lograr sus objetivos.

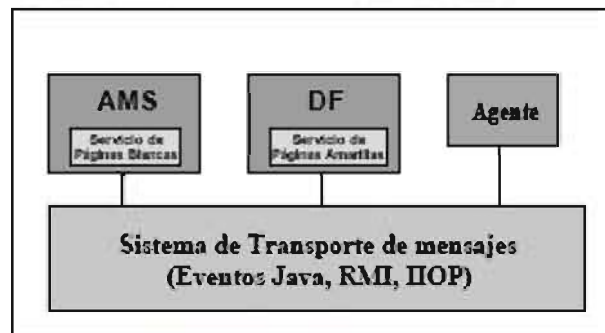


FIGURA 2.27. Arquitectura de JADE  
Fuente [PAV04]

### 2.6.2. PROGRAMACIÓN DE AGENTES EN JADE

JADE ofrece al desarrollador las siguientes funciones y ventajas:

- ◆ Ejecución distribuida.
- ◆ Interfaz gráfica para monitorización y depuración de agentes, incluyendo los que se encuentran en hosts remotos.
- ◆ Creación de agentes móviles.
- ◆ Ejecución de actividades en paralelo, según el paradigma de *behaviours*.
- ◆ Cumple con las especificaciones FIPA.
- ◆ Intercambio de mensajes ACL entre agentes.
- ◆ Registro automático de agentes (vía AMS).

- ◆ Servicio de nombres.

### **2.6.3. RAZONES PARA EL USO DE JADE**

Una pregunta que puede surgir es el por qué de usar JADE para desarrollar Sistemas Multi-Agente. JADE simplifica la comunicación y la cooperación entre los agentes, que tienen de forma distribuida la lógica de control para alcanzar el objetivo de la aplicación.

- ◆ Los agentes JADE pueden controlar su propio ciclo de vida, y pueden ser programados para que dejen de funcionar o empiecen a hacerlo dependiendo del estado del sistema y de la función que debe realizar el agente.
- ◆ JADE cumple con la especificación de FIPA, luego puede comunicarse con agentes realizados en otros entornos que sigan FIPA.
- ◆ Es código abierto. Multitud de personas colaboran en la realización y mantenimiento de JADE. La evolución de JADE es controlada por el JADE Governing Board, para que su crecimiento no se realice de forma desordenada.
- ◆ Los agentes JADE pueden correr en las diferentes versiones de Java: J2EE, J2SE y J2ME.
- ◆ El API proporcionado por JADE es intuitivo, fácil de aprender y sencillo de usar, haciendo que el desarrollo se produzca de manera mas rápida que si no se utilizase.

Por estas características, los principales campos de aplicación son:

- ◆ Aplicaciones móviles: Facilita la búsqueda y procesamiento de la información.
- ◆ Internet: Desarrollo de aplicaciones P2P.

## **CAPITULO III**

# **DESARROLLO DE SMA DE LICITACIONES**

### **3.1. DEFINICION CASO DE ESTUDIO SISTEMA MULTI-AGENTE DE LICITACIONES**

#### **3.1.1. LICITACION**

En la Licitación<sup>5</sup> intervendrán compradores (Ofertantes) y vendedores (Licitantes), el último recibe ofertas por artículos (todos los artículos son de interés para el o los ofertantes) puestos en oferta siguiendo alguna convención de oferta preestablecida que determina tanto la secuencia de ofertas, como la manera como se designa al ganador y el monto a-l que se adjudica el artículo. El tipo de Licitación desarrollado en el caso de estudio reúne las siguientes características:

- ◆ Primer precio, en sobre cerrado
- ◆ Oferta única
- ◆ El género se vende a aquel que ofrece el mayor precio
- ◆ Se paga el valor de la oferta más alta
- ◆ Estrategia dominante: Ofertar por debajo del valor propio.

#### **3.1.2. SISTEMA MULTI-AGENTE DE LICITACIONES**

El sistema se concibe como un Sistema Multi-Agente en el que el usuario es representado entre un agente conocido como el Agente Licitador y un conjunto de Agentes Ofertantes.

---

<sup>5</sup> Una Licitación según la Real Academia de la Lengua Española es ofrecer precio por una cosa (artículo, producto, etc.), es decir es un mecanismo para la fijación de precios en el que la negociación se basa a un proceso sumamente simple de coordinación. Generalmente la palabra Licitación es usada para describir una forma de contratación, usada por el Estado o Empresas Privadas para adquirir productos, servicios concesiones, etc. Para el contexto de estudio y objetivo de la presente tesis se utilizara la palabra Licitación como la definición que ofrece la Real Academia de la Lengua Española.

El objetivo de la Licitación es que el Agente Licitador asigne un artículo a uno de los Agentes Ofertantes. En muchos escenarios, el Agente Licitador desea maximizar el precio mientras que los Agentes Ofertantes desean minimizarlo.

### **3.1.3. NEGOCIACION**

Las Licitaciones solo se preocupan de la asignación de bienes: Requieren mejorar técnicas para lograr acuerdos.

La Negociación es el proceso que logra acuerdos en temas de interés común. Un escenario de negociación tiene cuatro componentes:

- ◆ Un conjunto de negociación ( propuestas posibles)
- ◆ Un protocolo.
- ◆ Estrategias privadas( una por cada agente)
- ◆ Una regla que determina cuando se ha logrado un trato y cuales el acuerdo.

#### **3.1.3.1 PROPUESTAS POSIBLES DE LA NEGOCIACION**

- ◆ Eficiente, optimo de Pareto<sup>6</sup>.
- ◆ Estable, no hay incentivo para desviarse
- ◆ Simple, bajo costo computacional y de comunicación
- ◆ Distribuida, no hay toma de decisiones centralizada
- ◆ Simétrica, agentes toman roles equivalentes.

#### **3.1.3.2. PROTOCOLO DE NEGOCIACION**

Las reglas del protocolo de negociación son:

- Se realizan varias rondas en la primera ronda, los agentes sugieren un trato simultáneamente acuerdo: cuando un agente detecta que el trato del otro es al menos tan bueno como el suyo si no hay acuerdo, se lanza otra ronda ningún agente puede proponer un trato peor para los demás que en la ronda anterior si ningún agente realiza una concesión, la negociación termina con el acuerdo conflicto. Se garantiza que el protocolo termina con un acuerdo (el número de tratos posibles es finito)

#### **3.1.3.3. ESTRATEGIA DE NEGOCIACION:**

Según la estrategia de Zeuthen<sup>7</sup>, los problemas a resolver por cada agente son:

- ◆ ¿Cuál debe ser la primera propuesta?

---

6. Aquella situación o circunstancia económica en la cual ninguna de las partes puede obtener un beneficio económico sin que la otra se vea perjudicada. En pocas palabras, para que éste óptimo se plasme, debe existir un equilibrio, donde no se puede dar ni pedir sin afectar el sistema económico.

7. Algoritmo de negociación sobre el conjunto de tratos puros individual-racionales en un análisis conceptual de las interacciones en dominios de múltiples agentes.



- ◆ ¿Quién debe ceder en cada ronda?
- ◆ ¿Cuánto se debe ceder?
- ◆ El acuerdo más favorable.
- ◆ El agente con menor riesgo de conflicto.
- ◆ Lo justo para salir del riesgo.

En base a todo lo mencionado en esta sección, y con las reglas claras del Sistema Multi-Agente de Licitaciones ahora es posible continuar con uno de los objetivos de la tesis que es el modelar el mismo SMA de Licitaciones con dos metodologías MAS-commonKADS e INGENIAS, a medida que se realice el modelado se especificaran algunos aspectos y detalles del SMA de Licitaciones que se solo se obtienen al realizar el modelado.

## 3.2. SMA DE LICITACIONES CON MAS-COMMONKADS

### 3.2.1. CONCEPTUALIZACION

En la Conceptualización se obtiene una primera descripción preliminar del problema y para lo cual se utilizan casos de uso que puedan ayudar a obtener requerimientos de información

**Documento 8**  
(Clave:Doc8)

**Descripción de actividades y Notación de Casos de Uso de la Fase Conceptualización**

#### ➤ DESCRIPCION DE ACTIVIDADES

##### a) Actividad: Identificación de los actores

EL SMA de Licitaciones es muy elemental solo interactúan dos tipos de agentes

- ◆ agente ofertante
- ◆ agente licitante.

##### b) Actividad: Descripción de los actores

Nombre **Agente ofertante**: encargado de realizar la oferta al iniciar una Licitación.

Nombre **Agente licitante**: encargado de iniciar y finalizar las Licitaciones, recibe y publica las ofertas.

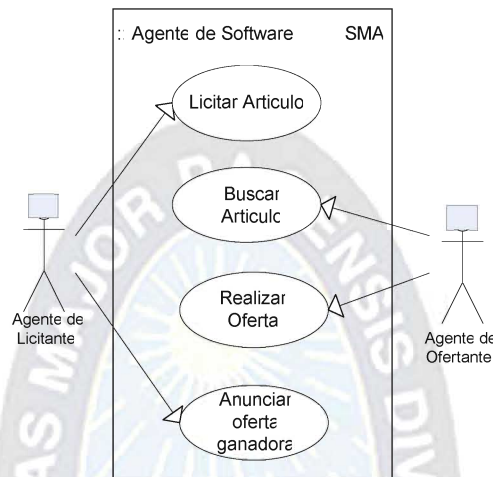
##### c) Actividad: Identificación de los casos de uso

Nuestro sistema cuenta con cuatro casos de uso.

- ◆ **Licitar Artículos**: Se incluyen los artículos a licitar y se publican con el agente licitante y el precio base.
- ◆ **Buscar Artículos**: Se realiza una búsqueda del artículo a adquirir al encontrarlo se comprueba que el precio base no sobrepase el precio tope, en el actual caso de uso todo artículo interesa a nuestros ofertantes.

- ◆ **Realizar oferta:** Se realiza la oferta por el artículo seleccionado.
- ◆ **Anunciar oferta ganadora:** Se publica la oferta ganadora hasta el momento para que los ofertantes realicen una oferta mayor, si no se obtiene una oferta mayor de cierto tiempo se declara como la oferta final y ganadora.

➤ **NOTACIÓN DE LOS CASOS DE USO**



**FIGURA 3.1.** Casos de Uso del SMA de Licitaciones.

Fuente [Elaboración Propia]

• **Caso de uso “Licitar Artículo”**

<b>Actores</b>	Licitante.
<b>Precondición</b>	Debe existir un artículo a licitar.
<b>Excepción</b>	No se incluyen artículos a licitar.
<b>Descripción</b>	Se toma el artículo seleccionado para licitar, se realiza un incremento de ganancia, y se publica en una pizarra para que los ofertantes se enteren de ella.

**Tabla 8.1.** Descripción de Caso de Uso- Licitar Artículo

Fuente [IGL98, Elaboración Propia]

• **Caso de uso “Buscar Artículos”**

<b>Actores</b>	Ofertante.
<b>Precondición</b>	Debe existir un artículo a licitar con su respectivo precio, y este no sobrepase el dinero disponible.
<b>Excepción</b>	No sé han incluido artículos a licitar. O el precio sobrepasa el monto disponible.
<b>Descripción</b>	Se identifica la presencia de un artículo y se anuncia el interés de participar en la Licitación.

**Tabla 8.2.** Descripción de Caso de Uso-Buscar Artículos

Fuente [IGL98, Elaboración Propia]

• **Caso de uso “Realizar Oferta”**

<b>Actores</b>	Ofertante.
<b>Precondición</b>	El precio del artículo para realizar la oferta aumentada no debe sobrepasar el monto disponible.
<b>Excepción</b>	El precio es muy alto para competir.
<b>Descripción</b>	El precio del artículo aumentado para realizar la oferta no sobrepase el monto disponible.

**Tabla 8.3.** Descripción de Caso de Uso-Realizar Oferta

Fuente [IGL98, Elaboración Propia]

• **Caso de uso “Anunciar Ganador”**

Actores	Licitante
Precondición	Se recibió la oferta de un ofertante, la cual es mayor a la oferta o ganadora
Excepción	
Descripción	Se publica la oferta ganadora hasta el momento para que los ofertantes realicen una oferta mayor, si no se obtiene una oferta mayor en cierto tiempo se declara como la oferta final y ganadora.

**Tabla 8.4.** Descripción de Caso de Uso-Anunciar Ganador  
Fuente [IGL98, Elaboración Propia]

**Tabla 8. Documento 8 (Fase de Conceptualización)**

### 3.2.2. ANALISIS

Con el fin de determinar los requerimientos del sistema iniciando así por establecer el problema se obtienen las especificaciones del SMA en la etapa de análisis corresponde a la segunda fase de la metodología de desarrollo.

El resultado es la especificación de requerimientos del Sistema Multi-Agente a través del desarrollo de todos los modelos a excepción del de diseño. Es decir los pasos son los siguientes:

- ◆ Modelado de agentes
- ◆ Modelado de tareas
- ◆ Modelado de la coordinación
- ◆ Modelado del experiencia
- ◆ Modelado de la organización

#### 3.2.2.1. MODELO DE AGENTES

Con los actores identificados en la fase anterior se identifica los agentes que denotan al SMA de Licitación. Cada actor en nuestro caso identifica un agente por lo cual tenemos dos agentes.

- Agente Ofertante.
- Agente Licitante.

**Documento 9  
(Clave:Doc9)**

**Modelo de Agente**

Los modelos de agentes se definen a través de las tarjetas CRC (Clase / Responsabilidad / Colaboración)

AGENTE LICITANTE		CLASE: SOFTWARE		
Objetivos	Planes	Conocimiento	Colaborador	Servicio
<b>Ofrecer Licitación</b>	Ofrecer artículo, con precio inicial.	Precio base del artículo		Ofrecer Licitaciones
<b>Decidir ganador</b>	Recibir ofertas e informar la mayor	Tiempo limite	Agente Ofertante	Terminar Licitación y

	hasta cumplir el tiempo limite.			decidir ganador.
--	---------------------------------	--	--	------------------

**Tabla 9.1.** Tarjeta CRC para Agente Licitante  
Fuente [IGL98, Elaboración Propia]

AGENTE OFERTANTE		CLASE: SOFTWARE		
Objetivos	Planes	Conocimiento	Colaborador	Servicio
<b>Identificar Licitación</b>	Reconocer Licitación de interés que no sobrepase el tope.	Precio base		Inscribirse e la Licitación.
<b>Ofertar</b>	Realizar oferta por el artículo.	Fórmula de incremento, estipulada, para ganar el artículo al realizar ofertas.		Ofertar

**Tabla 9.2.** Tarjeta CRC para Agente Ofertante  
Fuente [IGL98, Elaboración Propia]

### ➤ DESCRIPCION DE AGENTES

Una vez definidos los Agentes mediante las tarjetas CRC pasamos a describir las características del mismo lo cual será base para el Modelo de Experiencia, Conocimiento y Tareas. Consideraremos los parámetros como: el tipo, rol, descripción y objetivos.

#### a) Agente Licitante

**TIPO:** Agente de Software/ Clase Entidad

La entidad Agente Licitante se define como el que interactúa con el/los Agentes Ofertantes.

**ROL:** Vender

Vender es el rol que desempeña el agente Licitante, es también el papel que desempeña en el modelo de Organización.

#### **DESCRIPCION**

El Agente Licitante como una entidad capaz poner un producto a la venta con un precio base, después de seguir una serie de ofertas decidir que Agente Ofertante (comprador) gana y dar el nombre del ganador, dada cierta cantidad de tiempo.

#### **OBJETIVOS**

Vender el artículo de interés para el comprador en el precio más alto dado por el Agente Ofertante.

#### b) Agente Ofertante

**TIPO:** Agente de Software/ Clase Entidad

La entidad Agente Ofertante se define como el que interactúa con el Agente Licitante.

**ROL:** Comprar

Comprar es el rol que desempeña el agente Ofertante, es también el papel que desempeña en el modelo de Organización.

#### **DESCRIPCION**

El agente Ofertante inicializa con una cantidad determinada de dinero, realiza ofertas en base al tope del dinero que tiene.

**OBJETIVOS**

Comprar artefactos con la menor cantidad de dinero posible.

**Tabla 9. Documento 9 (Modelo de Agente)**

**3.2.2.2. MODELO DE TAREAS**

El modelo de tareas permite mostrar la descomposición funcional del sistema, y parte de las áreas funcionales (constituyente función) de la organización en que se va a introducir el sistema inteligente. La descripción de una tarea incluye:

- ◆ su nombre
- ◆ ingredientes de entrada y salida
- ◆ estructura de la tarea
- ◆ su control
- ◆ frecuencia de aplicación
- ◆ precondiciones, y
- ◆ capacidades requeridas para los realizadores.
- ◆ una descripción corta

En el presente caso de uso, el SMA no es muy inteligente.

**Documento 10 (Clave:Doc10) Modelo de Tareas**

Las tareas identificadas para el SMA de Licitaciones son las siguientes:



**FIGURA 3.2.** Tareas de SMA de Licitaciones. Fuente [Elaboración Propia]

➤ **DESCRIPCION DE TAREAS**

<b>Tarea</b>	Publicar Licitación
<b>Tipo</b>	Objetivo persistente.
<b>Parámetros de entrada</b>	Ingreso de tipo de artículo, precio de base
<b>Parámetros de salida</b>	Licitación, número de licitador, precio.
<b>Condición de activación</b>	Al nacer el agente y recibir el artículo

<b>Condición de finalización</b>	Al terminar la Licitación y vender el artículo <b>Condición de finalización.</b>
<b>Descripción</b>	El agente decide que Licitación es la más adecuada y publica el agente

**Tabla 10.1.** Descripción Tarea Publicar Licitación  
Fuente. [ROD07, Elaboración propia]

<b>Tarea</b>	Realizar oferta
<b>Tipo</b>	Objetivo persistente
<b>Parámetros de entrada</b>	Registro de Licitación, y precio base
<b>Parámetros de salida</b>	Oferta del artículo, con precio inferior al tope
<b>Condición de activación</b>	Al nacer el agente
<b>Condición de finalización</b>	Ganar Licitación, o sobrepasar el precio tope
<b>Descripción</b>	El agente ofertante realiza la ofertas necesarias hasta que se decrete un ganador o la oferta siguiente sobrepase el precio tope propio.

**Tabla 10.2.** Descripción Tarea Realizar Oferta  
Fuente. [ROD07, Elaboración propia]

<b>Tarea</b>	Decidir Ganador
<b>Tipo</b>	Objetivo persistente.
<b>Parámetros de entrada</b>	Ingreso de oferta más alta, tiempo limite.
<b>Parámetros de salida</b>	Oferta ganadora, numero de Ofertante, precio.
<b>Condición de activación</b>	Al nacer el agente y recibir el articulo.
<b>Condición de finalización</b>	Al terminar la Licitación y vender el articulo.
<b>Descripción</b>	El agente decide que Licitación es la ganadora y publica el agente licitante ganador.

**Tabla 10.3.** Descripción Tarea Decidir Ganador  
Fuente. [ROD07, Elaboración propia]

<b>Tarea</b>	Minimizar Precio
<b>Tipo</b>	Meta.
<b>Parámetros de entrada</b>	Precio de artefacto, precio tope.
<b>Parámetros de salida</b>	Ofertar bajo.
<b>Condición de activación</b>	Al nacer el agente y recibir precio del artículo.
<b>Condición de finalización</b>	Articulo vendido o tope de dinero.
<b>Descripción</b>	El agente minimiza los mas que puede el precio ofertando bajo.

**Tabla 10.4.** Descripción Tarea Minimizar Precio  
Fuente. [ROD07, Elaboración propia]

<b>Tarea</b>	Ofertar Bajo
<b>Tipo</b>	Meta
<b>Parámetros de entrada</b>	Precio base de artículo, precio tope.
<b>Parámetros de salida</b>	Precio de mayor oferta en base al precio tope.
<b>Condición de activación</b>	Al minimizar el precio
<b>Condición de finalización</b>	Al terminar la Licitación
<b>Descripción</b>	El agente realiza ofertas pequeñas en base al precio tope que contiene y los precios publicados en la pizarra.

**Tabla 10.5.** Descripción Tarea Ofertar Bajo  
Fuente. [ROD07, Elaboración propia]

<b>Tarea</b>	Maximizar Precio
<b>Tipo</b>	Objetivo persistente
<b>Parámetros de entrada</b>	Registro de Compradores y precio base de artefacto
<b>Parámetros de salida</b>	Subir Precio
<b>Condición de activación</b>	Al nacer el agente y al conocerse el precio base del artefacto
<b>Condición de finalización</b>	Precio máximo y tiempo limite
<b>Descripción</b>	El agente Licitante a medida que las ofertas llegan las publica en la pizarra, así los agentes ofertantes maximizan el precio del artefacto.

**Tabla 10.6.** Descripción Tarea Realizar Oferta  
Fuente. [ROD07, Elaboración propia]

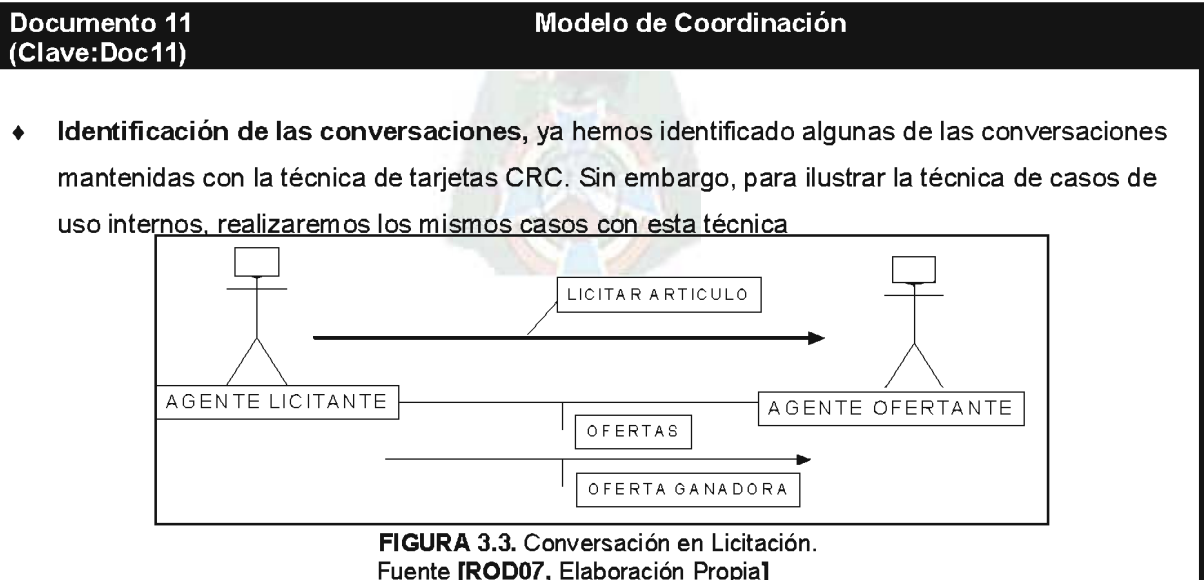
<b>Tarea</b>	Subir Precio
<b>Tipo</b>	Meta
<b>Parámetros de entrada</b>	precio base
<b>Parámetros de salida</b>	Precio mayor de venta de artefacto
<b>Condición de activación</b>	Al conocerse el precio base del artefacto
<b>Condición de finalización</b>	Máximo precio pagado por agentes ofertantes
<b>Descripción</b>	El agente ofertante realiza la ofertas necesarias hasta que se decrete un ganador o la oferta siguiente sobrepase el precio tope propio.

**Tabla 10.7.** Descripción Tarea Realizar Oferta  
Fuente. [ROD07, Elaboración propia]

**Tabla 10.** Documento 10 (Modelo de Tareas)

### 3.2.2.3. MODELO DE COORDINACIÓN

Tiene dos objetivos la primera la definición de los canales de comunicación y construcción de un prototipo. Y la segunda fase el análisis de las interacciones y definición de interacciones complejas. Con el modelo de coordinación se describe el esquema de comunicación del SMA; es decir, la conversación, los protocolos y los lenguajes asociados.



- ◆ **Descripción de las conversaciones**, la descripción de las conversaciones se realiza desde dos puntos de vista:

a) **Estructural**, qué fases tiene la conversación y qué intervenciones se dan en cada fase. La conversación de los agentes es lineal, solo se interesa en hacer llegar el mensaje correspondiente a su destino.

b) **Externo**, analizamos cuál es el objetivo de la conversación, sus pre y pos condiciones, y qué participantes hay.

<b>CONVERSACIÓN</b>	<i>LICITAR ARTICULO</i>
<b>Tipo</b>	Entrega-de-información.
<b>Objetivo</b>	Informar la apertura de Licitación.
<b>Agentes</b>	Licitante, ofertante (Grupo).
<b>Iniciador</b>	Licitante.
<b>Servicio</b>	Artículo a licitar.
<b>Precondición</b>	Existencia de un artículo a licitar.
<b>Pos condición</b>	Información ofrecida (de éxito o error).
<b>condición-terminación</b>	Inicio de oferta, o finalización de tiempo.
<b>tiempo-ejecución</b>	El tiempo de ejecución no debe superar un minuto, que es el tiempo máximo recomendado para el uso de recursos.
<b>Descripción</b>	El agente <i>Licitante anuncia</i> a todos los ofertantes el inicio de un artículo en la Licitación.

**Tabla 11.1.** Descripción Conversación Licitar Artículo  
Fuente. [ROD07, Elaboración propia]

<b>CONVERSACIÓN</b>	<i>OFERTA</i>
<b>Tipo</b>	Entrega-de-información.
<b>Objetivo</b>	Realizar una oferta por el artículo.
<b>Agentes</b>	Ofertante, Licitante
<b>Iniciador</b>	Ofertante
<b>Servicio</b>	Oferta(artículo, cantidad, ofertante)
<b>Precondición</b>	Conocimiento de inicio de Licitación.
<b>Pos condición</b>	Información de oferta mayor
<b>condición-terminación</b>	Envío del mensaje.
<b>tiempo-ejecución</b>	Inmediato.
<b>Descripción</b>	El agente ofertante anuncia al agente Licitante su oferta por el artículo.

**Tabla 11.2.** Descripción Conversación Oferta  
Fuente. [ROD07, Elaboración propia]

<b>CONVERSACIÓN</b>	<i>OFERTA GANADORA</i>
<b>Tipo</b>	Entrega-de-información.
<b>Objetivo</b>	Informar mayor oferta recibida (artículo, oferta, ofertante)
<b>Agentes</b>	Licitante, ofertante (Grupo).
<b>Iniciador</b>	Licitante.
<b>Precondición</b>	Recepción de oferta.



<b>Pos condición</b>	Finalización de Licitación.
<b>condición-terminación</b>	Finalización de tiempo.
<b>tiempo-ejecución</b>	El tiempo de ejecución no debe superar un minuto, que es el tiempo máximo recomendado.
<b>Descripción</b>	El agente <i>Licitante anuncia</i> a todos los ofertantes la oferta por el artículo en la Licitación.

**Tabla 11.3.** Descripción Conversación Oferta Ganadora  
Fuente. [ROD07, Elaboración propia]

♦ **Determinación de grupos de agentes**

Durante la identificación de las conversaciones, se detecta un grupo, denominado *ofertantes* que describimos a continuación, con la plantilla del modelo de agente:

<b>Grupo</b>	OFERTANTES
<b>Tipo</b>	Público
<b>Componentes</b>	Ofertante.
<b>Servicio-ofrecido</b>	Oferta
<b>Descripción</b>	Grupo formado por agentes que ofrecen ofertas. Este grupo es público y realiza las ofertas en la Licitación.

**Tabla 11.4.** Descripción De Grupo de Agentes Ofertantes  
Fuente. [ROD07, Elaboración propia]

♦ **Canales básicos de comunicación**

Cuando ya son desarrolladas las conversaciones, se puede determinar qué interacciones (mensajes) son válidas (permitidas) entre los agentes, y representar esta información en un diagrama resumen.



**FIGURA 3.4.** Canales De Comunicación  
Fuente [Elaboración Propia]

**Tabla 11.** Documento 11 (Modelo de Coordinación)

**3.2.2.4. MODELO DE EXPERIENCIA**

**Documento 12**  
**(Clave:Doc12)**

**Modelo de Experiencia**

Al ser tan sencillos los agentes, no son susceptibles a la Experiencia, pues estarán guiados por simples formulas matemáticas, para la toma de decisiones, en cuanto a precio a licitar, precio de oferta de oferta y demás.

Este maneja una ontología muy básica con lo cual su dominio es completo pues su entorno

cambiara en el tiempo

- ◆ Mecanismo de Aprendizaje “Vetado para este ejemplo pues no lo utiliza.”
- ◆ Mecanismo de Razonamiento “Racionamiento estrictamente matemático”

<b>Representación</b>	Reglas matemáticas
<b>Tipo</b>	De acuerdo al acaso
<b>Grado de Confiabilidad</b>	Al utilizar un operador randomico es impreciso.
<b>Esquema de procesamiento</b>	No cambia con el tiempo.

**Tabla 12.1.** Cuadro de Experiencia  
Fuente. [ROD07, Elaboración propia]

Los agentes saben que existen otros agentes a su alrededor pero solo interesa que el licitante conozca a los ofertantes y el ofertante a un Licitante.

**Tabla 12.** Documento 12 (Modelo de Experiencia)

### 3.2.2.5. MODELO DE ORGANIZACIÓN

**Documento 13**  
(Clave:Doc13)

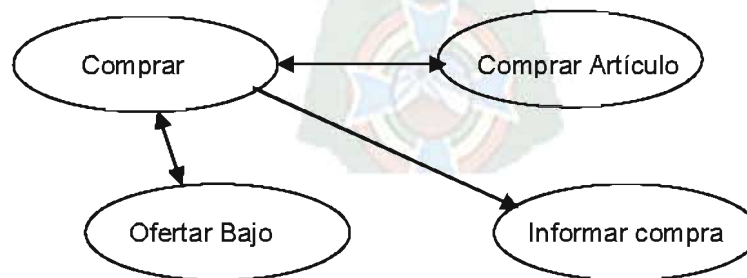
**Modelo de Organización**

El modelo de organización de MAS-CommonKADS tiene como objetivo analizar desde una perspectiva de grupo las relaciones entre los agentes (tanto software como humanos) que interactúan con el sistema.

- ◆ Por lo tanto es exclusividad a tratar aquí la relación agente soft- agente soft, pues no se evidencia interacción humana.
- ◆ La pizarra es una zona de trabajo común que permite a los agentes compartir todo tipo de información. Un SMA puede tener varias pizarras con distintos agentes registrados en cada una. No hay comunicación directa entre agentes.

#### NOTACION DE CASOS REACTIVOS

- ◆ **COMPRAR**



**FIGURA 3.5.** Caso reactivo para Comprar  
Fuente [Elaboración Propia]

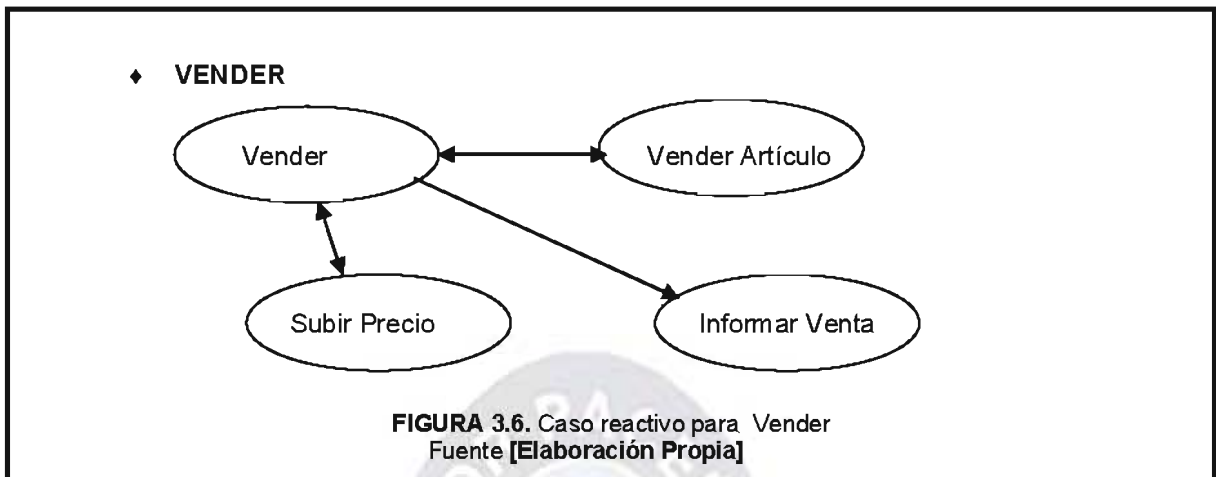


Tabla 13. Documento 13 (Modelo de Organización)

### 3.2.3. DISEÑO

Documento 14  
(Clave:Doc14)

Diseño de MAS-CommonKADS

#### 3.2.3.1. MODELO DE COMUNICACIÓN

El desarrollo del modelo de comunicación en un sistema multi-agente puede subdividirse en el conocimiento de la aplicación, el propio agente, el resto de agentes y el entorno. Detalla las interacciones entre los humanos y agentes de software, y los factores humanos para el desarrollo de las interfases de usuario.

El agente Licitante y el agente Ofertante no tienen capacidades reactivas, siendo así no podemos seleccionar una arquitectura reactiva ni híbrida, sino una deliberativa. Como se especificó en el Modelo de Organización la relación es agente soft- agente soft, pues no se evidencia interacción humana.

Es decir el módulo de comunicación con los agentes está integrado en la plataforma, y dado que Licitante y Ofertante no emplea ningún protocolo especial, no es necesario definirlo, ya que los mensajes intercambiados han sido ya expresados en FIFA.

#### 3.2.3.2. EL MODELO DE RED

El modelo de red proporciona a los agentes una visión uniforme de la red. Está definido como un modelo en capas, y distingue entre dos tipos de agentes:

El modelo de red en este ejemplo se omite pues todos trabajan sobre la misma plataforma JADE, que implementa paquetes de JADE la cual se rige al estándares de comunicación internacionales de la FIPA por lo tanto no hay necesidad de enfocarse en estos para este caso de uso que es sencillo.

Tabla 14. Documento 14 (Diseño de MAS-CommonKADS)

### 3.3. CASO DE ESTUDIO LICITACIONES CON INGENIAS

Para implementar el SMA de Licitaciones primero se realizara un esbozo de lo que se quiere realizar en el SMA. Para ello se identifican los casos de uso más importantes que reflejen los problemas principales que se encontraran y cuáles serán los componentes del sistema que participarán en su resolución.

Documento 15  
(Clave:Doc15)

Notación de Casos de Uso de Ingenias

Casos de uso asociados al problema

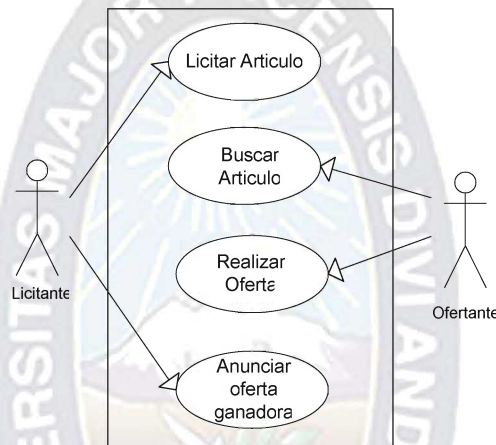


FIGURA 3.7. Caso de Uso para INGENIAS  
Fuente [Elaboración Propia]

- **Licitar Artículo:** Se incluyen los artículos a licitar y se publican con el agente licitante y el precio base.
- **Buscar Artículo:** Se realiza una búsqueda del artículo a adquirir al encontrarlo se comprueba que el precio base no sobrepase el precio tope, en el actual caso de uso todo artículo interesa a nuestros ofertantes.
- **Realizar oferta:** Se realiza la oferta por el artículo seleccionado.
- **Anunciar oferta ganadora:** Se publica la oferta ganadora hasta el momento para que los ofertantes realicen una oferta mayor, si no se obtiene una oferta mayor de cierto tiempo se declara como la oferta final y ganadora.

Tabla 15. Documento 15 (Notación de Casos de Uso de INGENIAS)

El SMA de Licitaciones ha sido desarrollado pensando solo en mostrar la funcionalidad del la metodología y la herramienta que provee IDK<sup>7</sup>. Aunque no indica todas características que pueden ser especificadas en un SMA, provee la suficientemente información para conseguir algunos agentes y desarrollarlos sobre una plataforma.

7. INGENIAS Development Kit (Herramienta de desarrollo de INGENIAS).

### 3.3.1. IMPLEMENTACION EN IDK

Documento 16 (Clave:Doc16)	Especificación de Diagramas
<p>En el desarrollo en IDK existen dos agentes, hay un Agente Licitante y otro Ofertante, para poder crear este SMA de Licitaciones el primer paso es identificar los diagramas en IDK:</p>	
<div style="border: 1px solid black; padding: 10px;"> <p>Specification Diagrams are:</p> <ul style="list-style-type: none"> <li>○ Project               <ul style="list-style-type: none"> <li>• Diagram name: <u>Agent Rol asociacion</u> type: AgentModel</li> <li>• Diagram name: <u>Objetivos y tareas</u> type: TasksAndGoalsModel</li> <li>• Diagram name: <u>modelo de interacción</u> type : InteractionModel</li> <li>• Diagram name: <u>flujo de evento</u> type : InteractionModel</li> <li>• Diagram name: <u>Ambiente</u> type : EnvironmentModel</li> </ul> </li> </ul> </div>	
<p><b>FIGURA 3.8.</b> Especificación de diagramas Fuente [Documento generado automáticamente con IDK 2.7]</p>	

Tabla16. Documento 16 (Especificación de diagramas de INGENIAS)

#### 3.3.1.1 MODELO DE AGENTE

El Modelo de Agente describe agentes particulares y los roles en que se encontrarán a lo largo de su vida en el sistema. En el SMA de Licitaciones el Agente Licitante tiene los artículos, y el Agente Ofertante tiene cuidado de encontrar y comprar los artículos, cada agente en el sistema tiene un papel o rol, como se especifica en el diagrama Agent Rol asociación es decir el Papel o Rol de Asociación del Agente, que es una diagrama de modelo de tipo Agente.

Documento 17 (Clave:Doc17)	Modelo de Agente
<div style="border: 1px solid black; padding: 10px;"> <p><b>Diagram name: Agent Rol asociacion</b> <b>Diagram type: AgentModel</b> <b>Description:</b></p> <pre> classDiagram     class Licitante     class Ofertante     class Vender     class Comprar     Licitante --&gt; Vender : &lt;&lt;WFPlays&gt;&gt;     Ofertante --&gt; Comprar : &lt;&lt;WFPlays&gt;&gt;           </pre> </div>	
<p><b>FIGURA 3.9.</b> Diagrama del Modelo de Agente Fuente [Documento generado automáticamente con IDK 2.7]</p>	

Se describe todas las entidades que aparecen en el Modelo de Agente, tomando en cuenta a Licitante, Ofertante, Vender y Comprar, con nombre, tipo y su respectiva descripción, como a continuación se muestra:

**Entities appearing in this diagram:**

NAME	TYPE	DESCRIPTION
Ofertante	Agent	It appears in diagrams: <ul style="list-style-type: none"> <li>• <u>Agent Rol asociacion</u>: AgentModel</li> <li>• <u>Objetivos y tareas</u>: TasksAndGoalsModel</li> <li>• <u>Ambiente</u>: EnvironmentModel</li> </ul>
Licitante	Agent	It appears in diagrams: <ul style="list-style-type: none"> <li>• <u>Agent Rol asociacion</u>: AgentModel</li> <li>• <u>Objetivos y tareas</u>: TasksAndGoalsModel</li> <li>• <u>Ambiente</u>: EnvironmentModel</li> </ul>
Vender	Role	It appears in diagrams: <ul style="list-style-type: none"> <li>• <u>Agent Rol asociacion</u>: AgentModel</li> <li>• <u>modelo de interaccion</u>: InteractionModel</li> <li>• <u>flujo de evento</u>: InteractionModel</li> </ul>
Comprar	Role	It appears in diagrams: <ul style="list-style-type: none"> <li>• <u>Agent Rol asociacion</u>: AgentModel</li> <li>• <u>modelo de interaccion</u>: InteractionModel</li> <li>• <u>flujo de evento</u>: InteractionModel</li> </ul>

**FIGURA 3.10.** Entidades del Modelo de Agente  
Fuente [Documento generado automáticamente con IDK 2.7]

A continuación se describen las relaciones de cada tipo de entidad y su fuente (*source*):

**Relationships appearing in this diagram:**

	TYPE	SOURCE
WFPlays	Ofertante	Comprar
WFPlays	Vender	Licitante
WFPlays	Vender	Licitante
WFPlays	Ofertante	Comprar

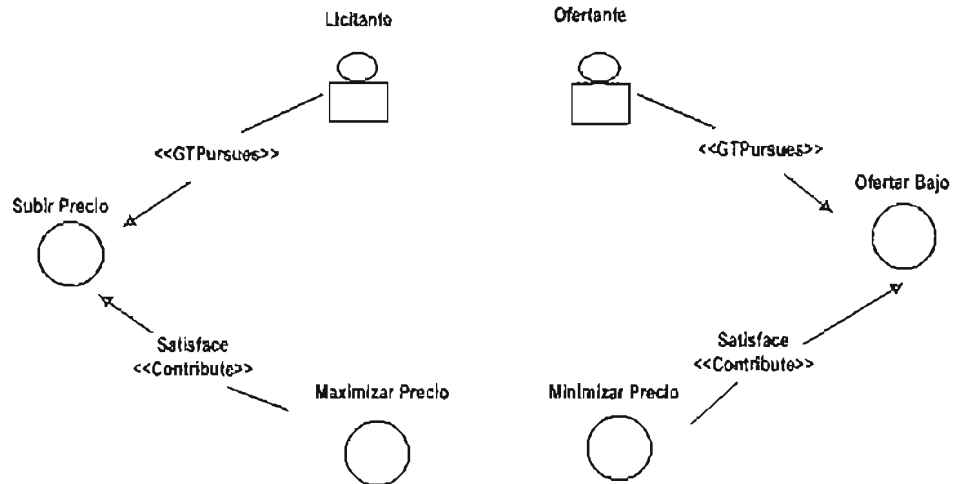
**FIGURA 3.11.** Relaciones que aparecen en el Modelo de Agente  
Fuente [Documento generado automáticamente con IDK2.7]

**Tabla 17. Documento 17 (Modelo de Agente de INGENIAS)**

### 3.3.1.2. MODELO DE OBJETIVOS Y TAREAS

El Modelo de Objetivos y Tareas se usa para asociar el estado mental del agente con las tareas que ejecuta. El estado mental puede verse como toda aquella información que permite al agente tomar decisiones. En otras palabras el comportamiento es definido por los objetivos y las tareas de los Agentes, y las interacciones entre agentes.

**Diagram name: Objetivos y tareas**  
**Diagram type: TasksAndGoalsModel**  
**Description:**



**FIGURA 3.12.** Diagrama del Modelo de Objetivos y tareas  
Fuente [Documento generado automáticamente con IDK 2.7]

El Ofertante ejerce el objetivo de buscar los artículos a comprar, con el compromiso de realizar las ofertas más bajas para conseguir el objetivo de minimizar precio, por el contrario el Agente Licitante tiene como compromiso maximizar precio, y vender el artículo.

Al crear el diagrama del Modelo de Objetivos y Tareas se describe las entidades que aparecen en el mismo, como a continuación se muestra:

**Entities appearing in this diagram:**

NAME	TYPE	DESCRIPTION
Maximizar precio	Goal	It appears in diagrams: <ul style="list-style-type: none"> <li>Objetivos y tareas: TasksAndGoalsModel</li> </ul>
Ofertar bajo	Goal	It appears in diagrams: <ul style="list-style-type: none"> <li>Objetivos y tareas: TasksAndGoalsModel</li> <li>flujo de evento: InteractionModel</li> </ul>
Subir precio	Goal	It appears in diagrams: <ul style="list-style-type: none"> <li>Agent Rol asociacion: AgentModel</li> <li>Objetivos y tareas: TasksAndGoalsModel</li> <li>Ambiente: EnvironmentModel</li> </ul>

<b>Licitacion</b>	<b>Agent</b>	It appears in diagrams: <ul style="list-style-type: none"> <li>• <a href="#">Agent Rol asociacion</a>: AgentModel</li> <li>• <a href="#">Objetivos y tareas</a>: TasksAndGoalsModel</li> <li>• <a href="#">Ambiente</a>: EnvironmentModel</li> </ul>
<b>Minimizar precio</b>	<b>Goal</b>	It appears in diagrams: <ul style="list-style-type: none"> <li>• <a href="#">Objetivos y tareas</a>: TasksAndGoalsModel</li> </ul>

**FIGURA 3.13.** Entidades que aparecen en el Modelo de Objetivos y Tareas  
Fuente [Documento generado automáticamente con IDK 2.7]

También se describen las relaciones de cada tipo de entidad y su fuente (*source*):

**Relationships appearing in this diagram:**

	TYPE	SOURCE
Contribute	Subir precio	Maximizar precio
GTPursues	Ofertante	Ofertar bajo
Contribute	Ofertar bajo	Minimizar precio
GTPursues	Subir precio	Licitacion
Contribute	Subir precio	Maximizar precio
GTPursues	Ofertante	Ofertar bajo
GTPursues	Subir precio	Licitacion
Contribute	Ofertar bajo	Minimizar precio

**FIGURA 3.14.** Relaciones que aparecen en el Modelo de Objetivos y Tareas  
Fuente [Documento generado automáticamente con IDK 2.7]

**Tabla 18. Documento 18 (Modelo de Objetivos y Tareas de INGENIAS)**

### 3.3.1.3. MODELO DE INTERACCION

El modelo de Interacción detalla como se coordinan y comunican los agentes, en este ejemplo hay solamente una interacción, que es mostrado. En INGENIAS una interacción podría ser una asociación multipartidista, con un iniciador y uno o más colaboradores, como en este ejemplo. A decir verdad, una interacción en INGENIAS es considerada algo más que sólo un mensaje - paso y podría representar mayor cantidad que la idea de una transacción, que puede estar constituido por el intercambio de varios envía un mensaje o notas en un espacio tuple compartido, dependiendo del paradigma de comunicación.

**Documento 19**  
**(Clave:Doc19)**

**Modelo de Interacciones**

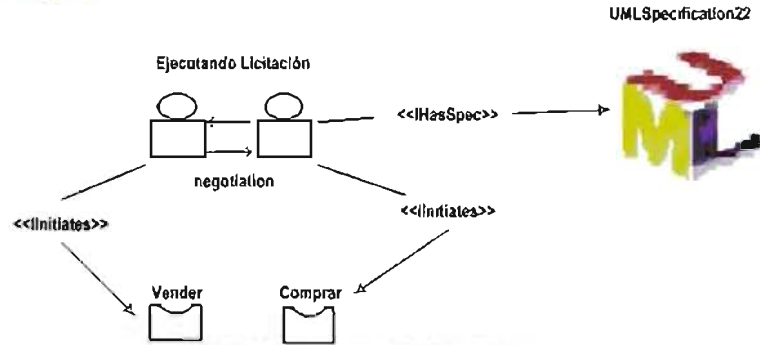
La circulación de los mensajes en la especificación puede ser especificado por un diagrama de colaboración de Gracia, o por otro clases de interacción las diagramas, como UML o diagramas de secuencia de UML de Agente, en INGENIAS las interacciones son provocadas en las tareas. La circulación de las tareas y las interacciones condiciona el comportamiento mundial del sistema. Esta información es muy importante cuando se genera las claves.



Diagram name: **modelo de interaccion**

Diagram type: **InteractionModel**

Description:



**FIGURA 3.15.** Diagrama del Modelo de Interacción  
Fuente [Documento generado automáticamente con IDK 2.7]

Las entidades y relaciones identificadas en el Modelo de Interacción son:

Entities appearing in this diagram :

NAME	TYPE	DESCRIPTION
Vender	Role	It appears in diagrams: <ul style="list-style-type: none"> <li>• <a href="#">Agent Rol asociacion</a>: AgentModel</li> <li>• <a href="#">modelo de interaccion</a>: InteractionModel</li> <li>• <a href="#">flujo de evento</a>: InteractionModel</li> </ul>
UMLSpecification22	UMLSpecification	It appears in diagrams: <ul style="list-style-type: none"> <li>• <a href="#">modelo de interaccion</a>: InteractionModel</li> <li>• <a href="#">flujo de evento</a>: InteractionModel</li> </ul>
Ejecutando Licitacion	Interaction	It appears in diagrams: <ul style="list-style-type: none"> <li>• <a href="#">modelo de interaccion</a>: InteractionModel</li> </ul>
Comprar	Role	It appears in diagrams: <ul style="list-style-type: none"> <li>• <a href="#">Agent Rol asociacion</a>: AgentModel</li> <li>• <a href="#">modelo de interaccion</a>: InteractionModel</li> <li>• <a href="#">flujo de evento</a>: InteractionModel</li> </ul>

**FIGURA 3.16.** Entidades que aparecen en el Modelo de Interacción  
Fuente [Documento generado automáticamente con IDK 2.7]

Relationships appearing in this diagram:

	TYPE	SOURCE
Initiates	Ejecutando Licitacion	Vender
IHasSpec	Ejecutando Licitacion	UMLSpecification22
IHasSpec	Ejecutando Licitacion	UMLSpecification22
Initiates	Ejecutando Licitacion	Vender
Initiates	Ejecutando Licitacion	Comprar
Initiates	Ejecutando Licitacion	Comprar

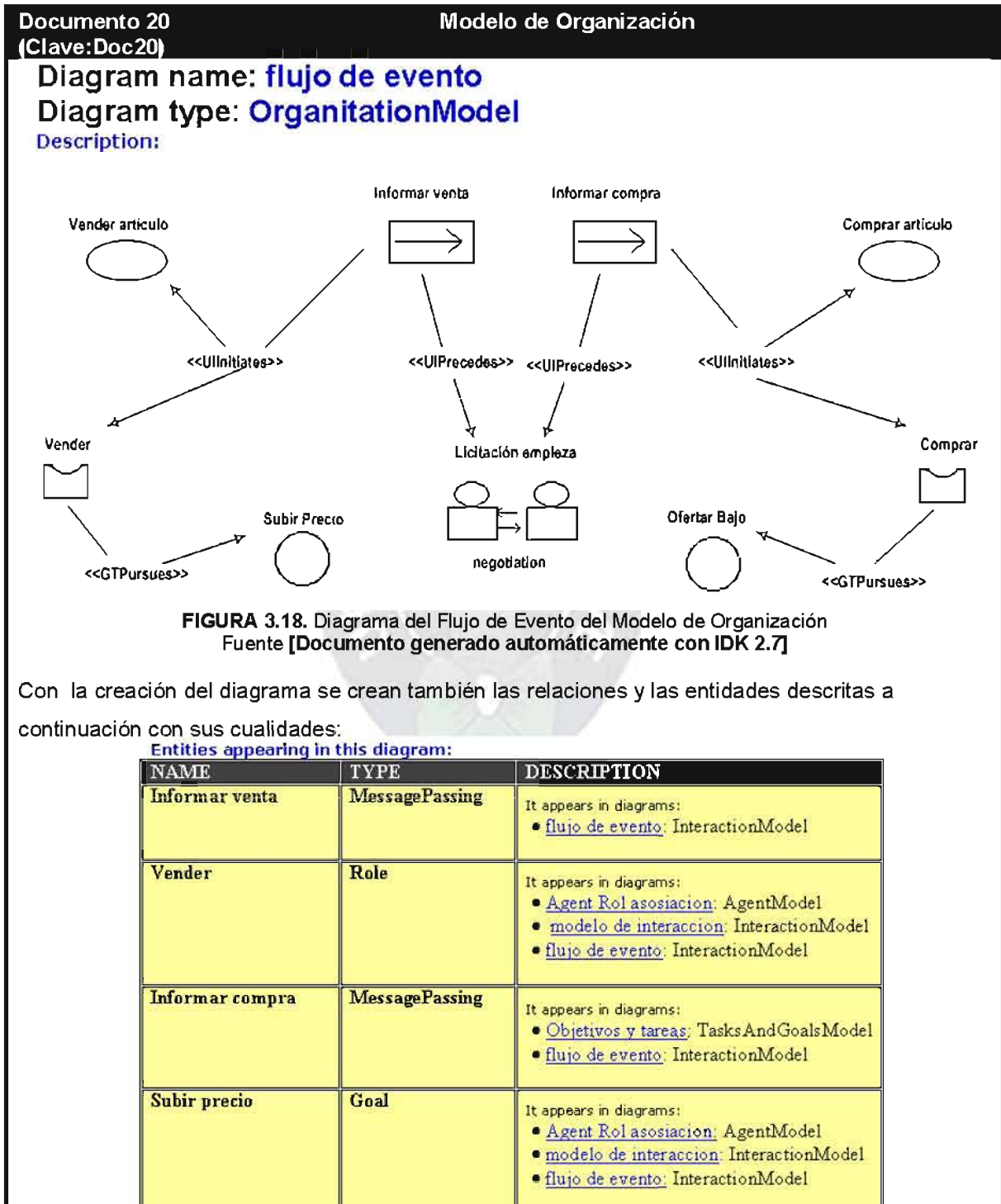
**FIGURA 3.17.** Relaciones que aparecen en el Modelo de Interacción  
Fuente [Documento generado automáticamente con IDK 2.7]

Tabla 19. Documento 19 (Modelo de Interacciones de INGENIAS)

### 3.3.1.4. MODELO DE ORGANIZACION

El Modelo de Organización define cómo se agrupan los agentes, la funcionalidad del sistema y qué restricciones hay que imponer sobre el comportamiento de los agentes.

En el siguiente diagrama se observa el comportamiento, la funcionalidad, es decir cual es el flujo de eventos posibles durante la interacción del SMA.



Comprar articulo	Task	It appears in diagrams: • <a href="#">flujo de evento</a> : InteractionModel
Ofertar bajo	Goal	It appears in diagrams: • <a href="#">Objetivos y tareas</a> : TasksAndGoalsModel • <a href="#">flujo de evento</a> : InteractionModel
Licitacion empieza	Interaction	It appears in diagrams: • <a href="#">flujo de evento</a> : InteractionModel
Vender articulo	Task	It appears in diagrams: • <a href="#">flujo de evento</a> : InteractionModel

**FIGURA 3.19.** Entidades que aparecen en el Flujo de Eventos  
Fuente [Documento generado automáticamente con IDK 2.7]

Relationships appearing in this diagram:

	TYPE	SOURCE	TARGET
UIPrecedes	Licitacion empieza	Informar venta	
UIInitiates	vender articulo	Vender	Informar venta
GTPursues	Subir precio	Vender	
UIInitiates	vender articulo	Vender	Informar venta
UIInitiates	Comprar	Comprar articulo	Informar compra
GTPursues	Comprar	Ofertar bajo	
UIInitiates	Comprar	Comprar articulo	Informar compra
UIPrecedes	Informar compra	Licitacion empieza	
GTPursues	Subir precio	Vender	
UIInitiates	Comprar	Comprar articulo	Informar compra
GTPursues	Comprar	Ofertar bajo	
UIPrecedes	Informar compra	Licitacion empieza	
UIPrecedes	Licitacion empieza	Informar venta	
UIInitiates	vender articulo	Vender	Informar venta

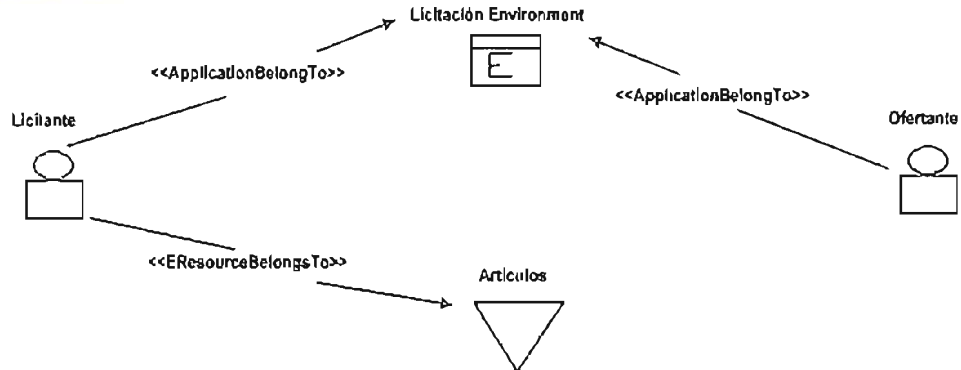
**FIGURA 3.20.** Relaciones que aparecen en el Flujo de Evento  
Fuente [Documento generado automáticamente con IDK 2.7]

**Tabla 20. Documento 20 (Modelo de Organización de INGENIAS)**

### 3.3.1.5. MODELO DE ENTORNO

El Modelo de Entorno define qué existe alrededor del nuevo sistema y cómo lo percibe cada agente. Así que los agentes creados para este ejemplo Agente Ofertante y Agente Licitante, residen en el mismo ambiente, y esto es representado por un diagrama de ambiente.

Diagram name: **Ambiente**  
Diagram type: **EnvironmentModel**  
Description:



**FIGURA 3.21.** Especificación del Ambiente del Modelo  
Fuente [Documento generado automáticamente con IDK 2.7]

Entidades y relaciones identificadas en el Modelo de Entorno:

Entities appearing in this diagram :

NAME	TYPE	DESCRIPTION
Licitacion Environment	Environment Application	It appears in diagrams: • <a href="#">Ambiente</a> : EnvironmentModel
Ofertante	Agent	It appears in diagrams: • <a href="#">Agent Rol asociacion</a> : AgentModel • <a href="#">Objetivos y tareas</a> : TasksAndGoalsModel • <a href="#">Ambiente</a> : EnvironmentModel
Articulos	Resource	It appears in diagrams: • <a href="#">Ambiente</a> : EnvironmentModel
Licitacion	Agent	It appears in diagrams: • <a href="#">Agent Rol asociacion</a> : AgentModel • <a href="#">Objetivos y tareas</a> : TasksAndGoalsModel

**FIGURA 3.22.** Entidades que aparecen en el Ambiente del Modelo  
Fuente [Documento generado automáticamente con IDK 2.7]

Relationships appearing in this diagram:

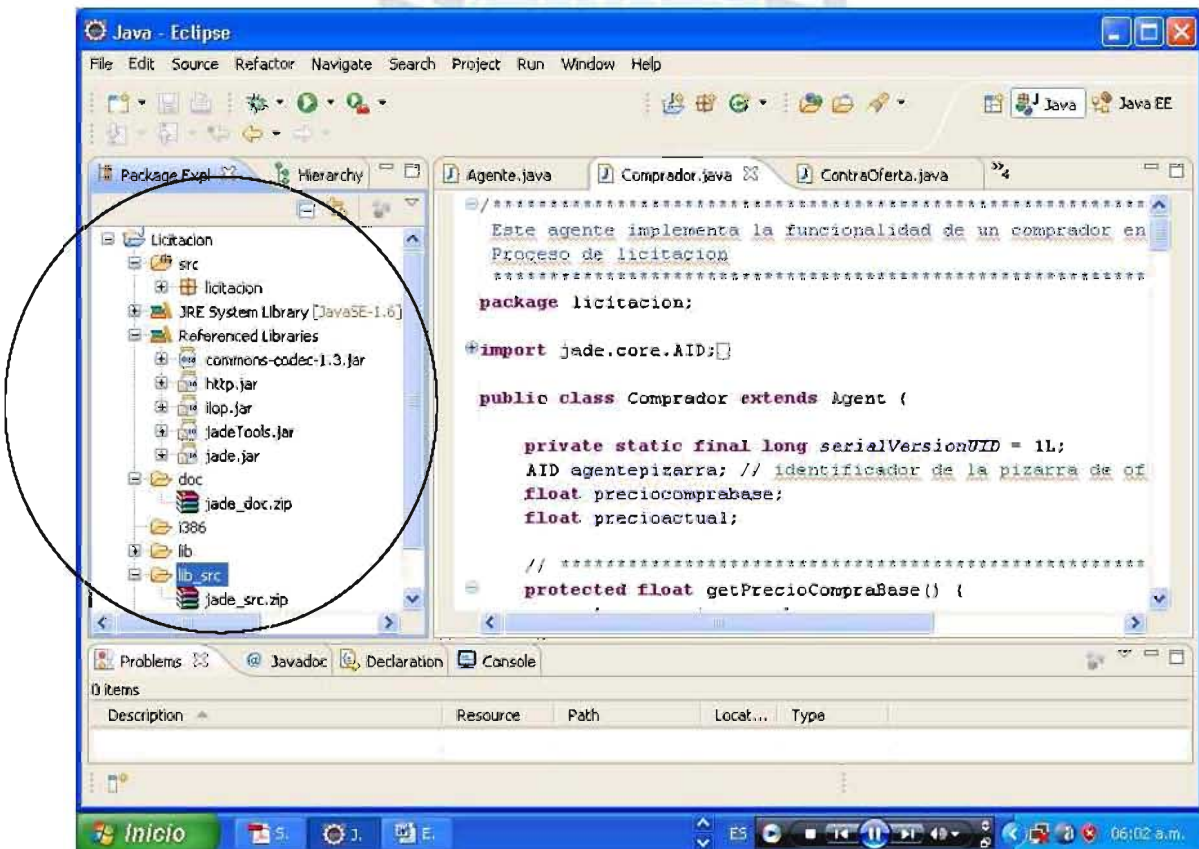
	TYPE	SOURCE
ApplicationBelongsTo	Licitacion Environment	Licitante
ApplicationBelongsTo	Ofertante	Licitacion Environment
ApplicationBelongsTo	Ofertante	Licitacion Environment
EResourceBelongsTo	Articulos	Licitante
EResourceBelongsTo	Articulos	Licitante
ApplicationBelongsTo	Licitacion Environment	Licitante

**FIGURA 3.23.** Relaciones que aparecen en el Ambiente del Modelo  
Fuente [Documento generado automáticamente con IDK 2.7]

### 3.4. CASO DE ESTUDIO LICITACIONES IMPLEMENTADO EN JADE

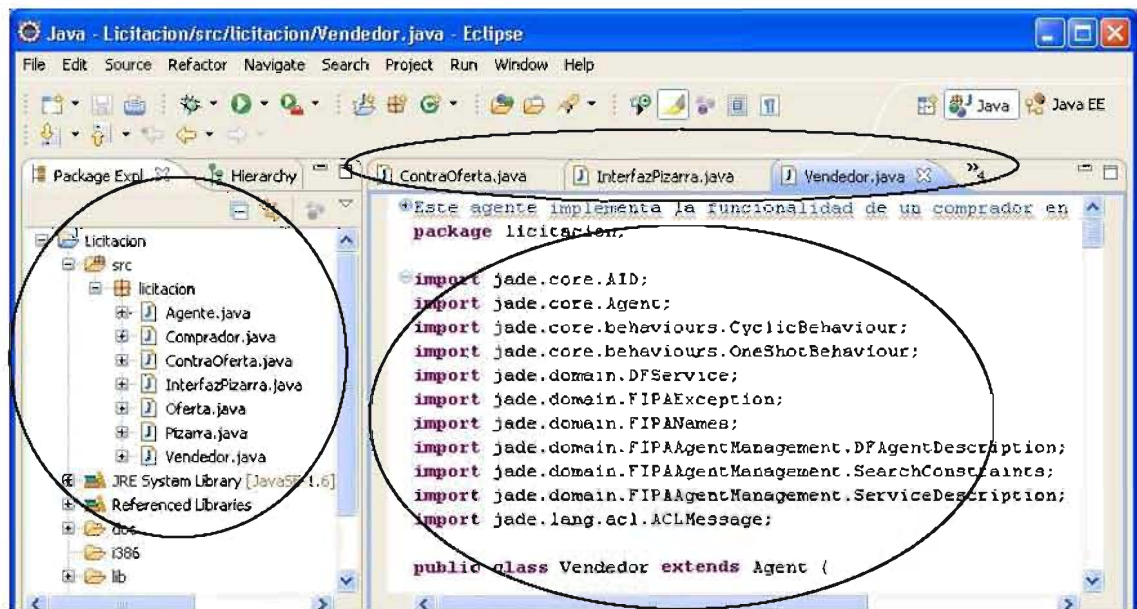
Al ser JADE una plataforma desarrollada en Java será posible utilizarla en cualquier sistema operativo que disponga de una máquina virtual de Java (la versión mínima es la 1.2). La página del proyecto la podemos encontrar en [\[WEBJADE\]](#), es desde esta página, en la sección *Download*, desde donde podremos obtener el archivo .zip de instalación. Podremos acceder a cuatro archivos con extensión .zip que se corresponderán con cada uno de los elementos de la plataforma (binarios, documentación, ejemplos, fuentes) o podremos obtener el archivo *JADE-all-version.zip*, que contendrá todos ellos. Se descomprime los zip, y lo siguiente es añadir a la variable de entorno CLASSPATH los archivos .jar del directorio *lib*, así como el *directorio actual*. Otra manera es utilizando Eclipse donde creamos un proyecto con los siguientes parámetros:

Si observamos la parte de los paquetes de Eclipse Figura 3.24., el proyecto se llama Licitación, en la carpeta de *src* se encuentra las clases que contiene licitación (ver Figura 3.25.), en “**Referenced Libraries**” es la carpeta donde exportamos los jars de JADE



**FIGURA 3.24.** Proyecto Licitación  
Fuente [Eclipse, Elaboración Propia]

(commos-codec-1.3.jar, http.jar, iiop.jar, jadeTools.jar, jade.jar), así ya no tenemos que configurar el CLASPATH. De la misma manera en la carpeta doc exportamos la documentación de JADE (jade\_doc.zip) y en lib\_src se exportan las librerías de JADE (jade\_src.zip).



**FIGURA 3.25.** Clases del Proyecto Licitación  
Fuente [Eclipse, Elaboración Propia]

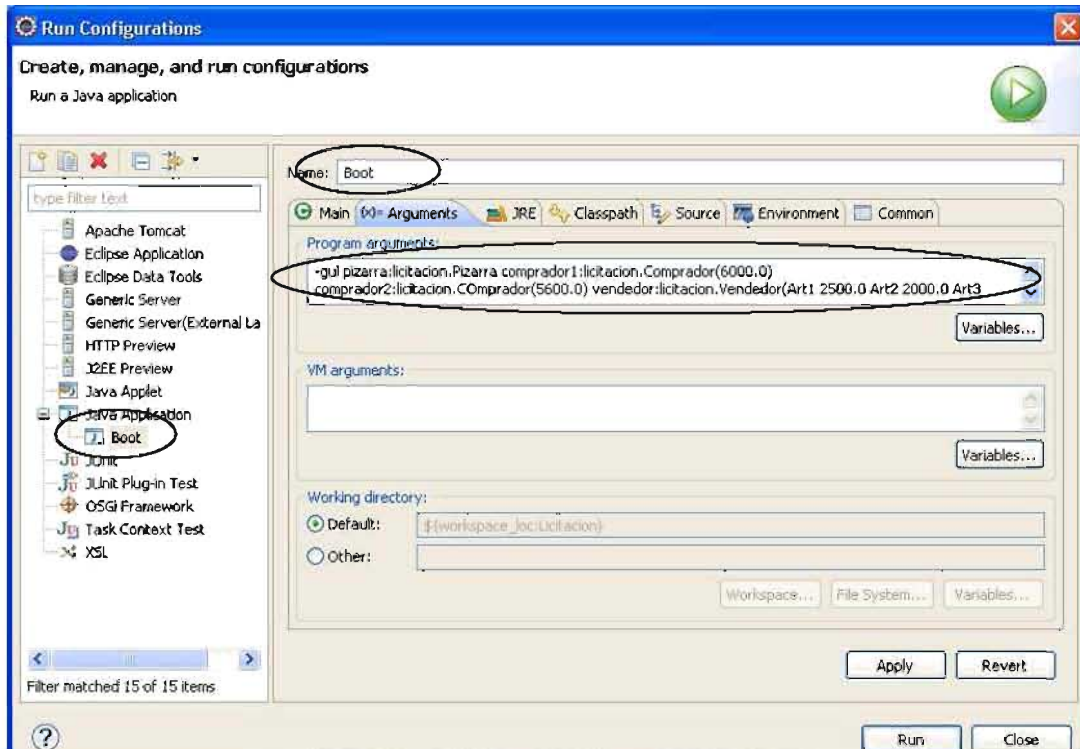
Las clases que contiene el proyecto Licitacion son 7: Agente.java, Comprador.java, Contraoferta.java, InterfazPizarra.java, Oferta.java, Pizarra.java y Vendedor.java.

Las clases Comprador.java, Vendedor.java y Pizarra.java heredan atributos de la clase Agente.java, además las tres clases mencionadas son las que utilizan las librerías de JADE, como observamos en la Figura 3.25. la clase Vendedor.java, para su comunicación con los estándares de ACL FIPA.

El programa fue diseñado para recibir los siguientes parámetros de entrada:

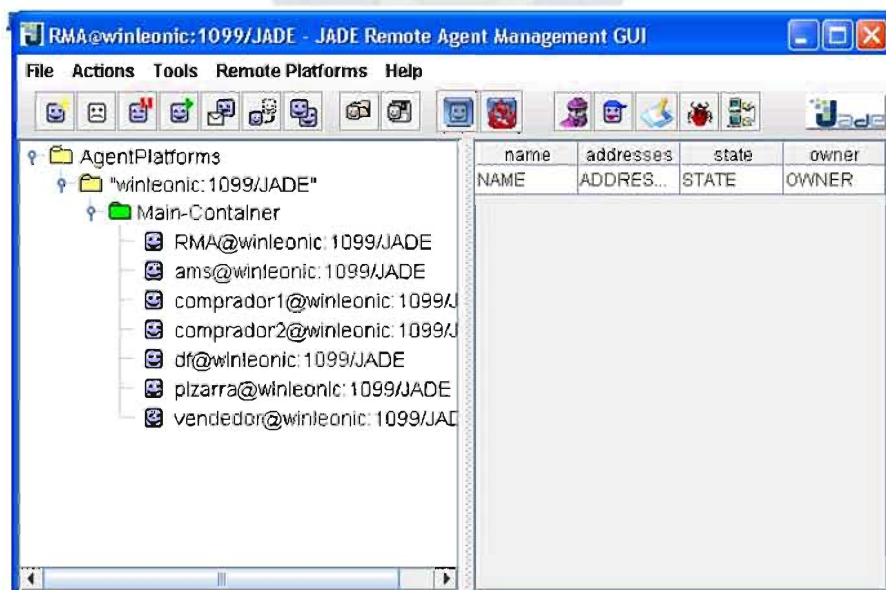
- ♦ -gui pizarra:licitacion.Pizarra, para llamar a la Interfaz grafica de Licitación, pizarra es el agente que pertenece al paquete licitación.
- ♦ Identificador\_de\_Comprador:licitacion.Comprador(cantidad\_de\_dinero\_disponible). El Identificador\_de Comprador es el nombre que se le da Comprador, puede existir uno o más compradores. Y Comprador es parte del paquete licitación.
- ♦ Identificador\_de\_vendedor:licitacion.Vendedor(Articulo1 PrecioArticulo1 Articulo2 PrecioArticulo2.....). Solo existe un vendedor y n artículos.

Los parámetros de entrada son configurados en los argumentos del Java Application escogiendo el Boot de Jade, para llamar a JADE. Y ponemos a correr el programa.



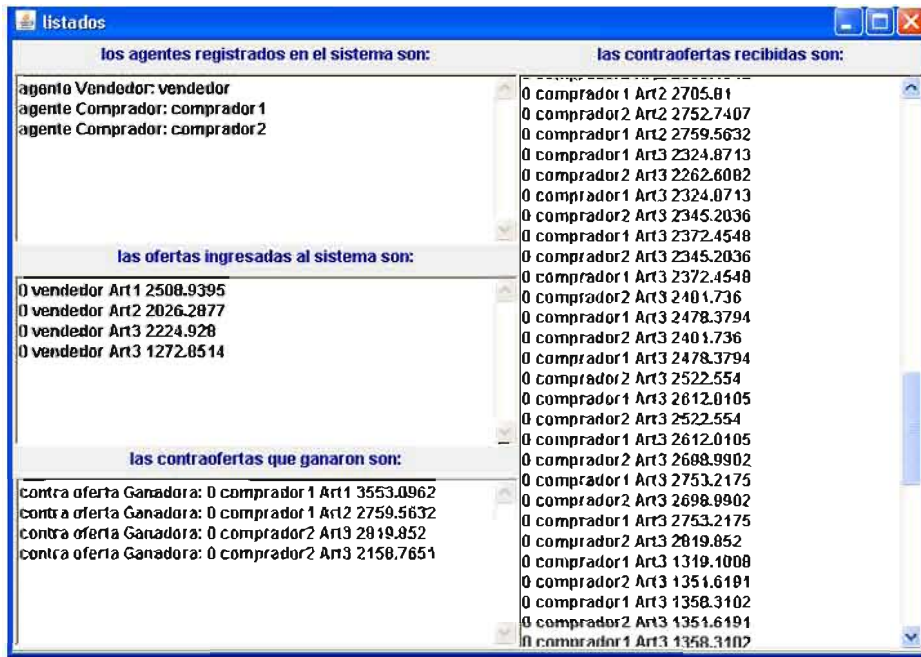
**FIGURA 3.26.** Configuración de Parámetros de entrada  
Fuente [Eclipse, Elaboración Propia]

El programa después de ser ejecutado llama a la plataforma JADE mostrando su modo grafico, donde se registran los agentes de tipo comprador, pizarra y vendedor.



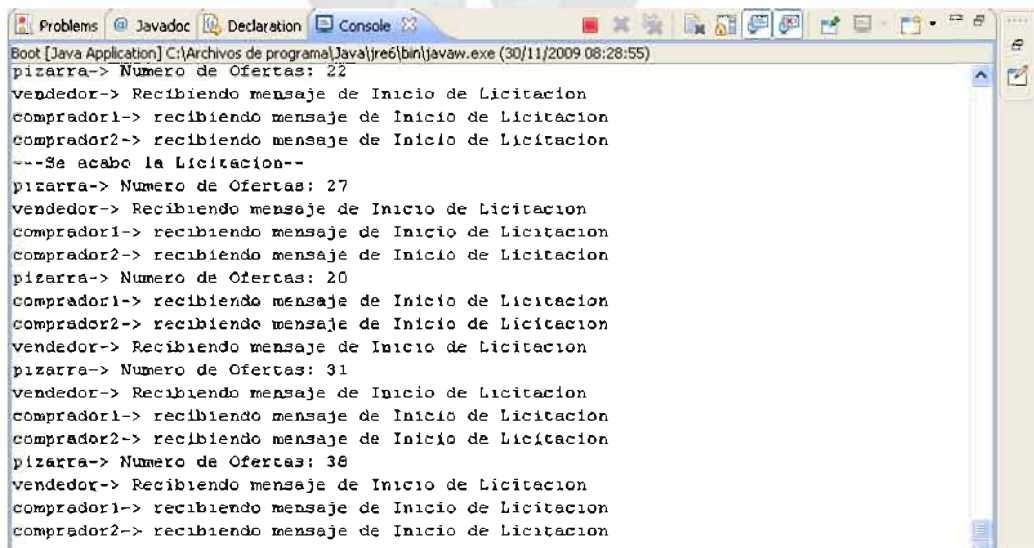
**FIGURA 3.27.** Clases del Proyecto Licitación  
Fuente [Eclipse, Elaboración Propia]

El modo grafico del programa desarrollado en java es la pizarra. Donde se observa a los Agentes “Registrados en el Sistema”, que “Ofertas ingresadas al Sistema” (los artículos a la venta), las “Contraofertas Ganadoras”, y las “Contraofertas Recibidas”.



**FIGURA 3.28.** Pizarra de Licitación  
Fuente [Eclipse, Elaboración Propia]

A continuación se muestra lo que ocurre en la consola de Eclipse a medida que el programa se ejecuta. Cada venta se realiza con un tiempo limite de 20 segundos, así que mientras mas artículos existan mas tiempo se requerirá para finalizar la Licitación



**FIGURA 3.29.** Consola de Eclipse en la Ejecución de Licitación  
Fuente [Eclipse, Elaboración Propia]



# CAPITULO IV

## EVALUACION COMPARACION DE METODOLOGIAS

### 4.1. CRITERIOS EVALUATIVOS Y COMPARATIVOS

Hasta la fecha, ningún estudio ha identificado una base "Estándar" de los pasos para la evaluación de una metodología. Investigar el proceso de desarrollo de las metodologías de AOSE<sup>9</sup> existentes se hace muy tedioso y difícil de hacer.

Para realizar la comparación de metodologías se extrajeron los peldaños más importantes de cada proceso de desarrollo y se los sintetiza en una lista coherente, esto lo hicieron los investigadores Quynh Nhu Numi Tran y Graham C. Bajo, ambos de la Universidad de Nueva Gales del Sur de Australia. Para verificar la fuerza legal de los pasos recomendados en [HEGI05], se dirigió una investigación de una visión general de expertos en cuanto a sus opiniones sobre la importancia de estos pasos. La encuesta confirmó la importancia de todos los pasos propuestos para el desarrollo de SMA, se considero hasta el último constituyente de los aspectos de las metodologías orientadas a SMA y también aportaron algunos criterios de evaluación los que todavía no han sido incluido en estructuras existentes como se "La metodología se acerca al desarrollo SMA", "El soporte para agentes móviles", y "Soporte para ontología"; también fue dirigida para probar la relación de los criterios propuestos y hacer las mejoras necesarias a la especificación de criterios.

Los evaluadores pueden usar condiciones diferentes y/o descripciones de los pasos metodología de AOSE, fue considerado aconsejable suministrar una base de "Estándar"

---

9. *Agent-Oriented Software Engineering* (Ingeniería de software Orientada a Agentes)

del proceso los pasos Ciclo de vida del desarrollo de un SMA.

#### 4.1.1 LA BASE DE EVALUACION

La estructura de evaluación seleccionada fue moldeada para identificar e integrar la evaluación de criterios basados en varias características del análisis de metodologías de desarrollo de sistema convencionales concretamente Wood, Pethia, Gold, y Firth (1988). La estructura de la base es mostrada en la Figura 3.23. Los criterios son agrupados en cuatro categorías:

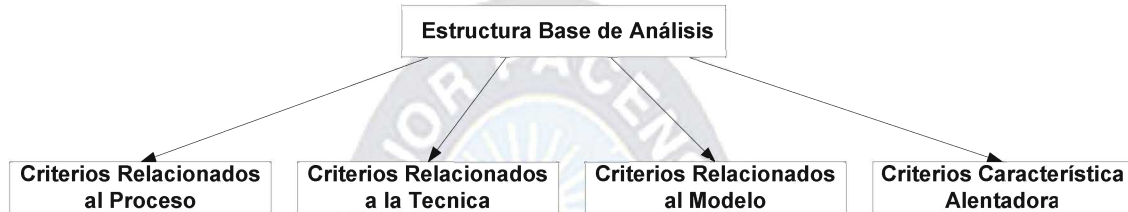


FIGURA 4.1. Estructura de la base de evaluación  
Fuente [HEGI05]

- **Criterios Relacionados al Proceso**, valorar el proceso de desarrollo de una metodología de AOSE.
- **Criterios Relacionados a la Técnica**, valorar la técnica de la metodología, llevar a cabo los pasos de desarrollo y/o producir modelos y componentes de notacionales.
- **Criterios Relacionados al Modelo**, revisar los modelos y componentes de notacionales de la capacidad y las características de la metodología.
- **Criterios Característica Alentadora**, valorar la variedad de capacidad metodológica de alto nivel.

Esta estructura es seleccionada en base a tres componentes en el desarrollo de Sistema Multi-Agente como el proceso, los modelos, y la técnica. Cada criterio de evaluación es acompañado por una o mas preguntas para controlar y ayudar para su evaluación.

#### 4.1.2. CRITERIOS RELACIONADOS AL PROCESO

Los criterios relacionados al proceso observan la aplicabilidad de la metodología, los pasos suministrados para su proceso de desarrollo, y el enfoque de desarrollo seguido por la metodología. Provee una lista bien arraigada de las características de ingeniería de sistema genéricas que son:

Documento 22 (Clave:Doc22)	Criterios Relacionados Al Proceso
a) <i>Desarrollo de Ciclo de vida</i>	¿Que Ciclo de vida de desarrollo describe mejor la metodología (Ej., Cascada o iterativo)?

<b>b) La cobertura del Ciclo de vida</b>	¿Que fases del Ciclo de vida serán cubiertos por la metodología (por ejemplo., El análisis, el diseño, y la puesta en práctica)?
<b>c) Perspectiva de Desarrollo</b>	¿Qué perspectivas de desarrollo soporta la metodología (Ej. Top-down, Bottom-Up o Hibrida)?
<b>d) Dominio de aplicación</b>	¿La metodología es aplicable a cualquier dominio de aplicación (Ej., Dominio independiente) o a un dominio específico (Ej., Dominio dependiente)?
<b>e) Tamaño de SMA</b>	¿Qué tamaño de SMA es apropiada para la metodología (Ej., SMA pequeño, o grande)?
<b>f) Naturaleza del Agente</b>	¿La metodología soporta cualquier arquitectura de Agentes, estimando el mecanismo, y/o la representación de conocimientos (Ej., Agentes heterogéneos), o solamente agentes de un tipo especial (Ej., Agentes homogéneos)?
<b>g) Soporte para comprobación y validación</b>	¿El proceso de desarrollo de la metodología contiene reglas para permitir la comprobación y la validación de la corrección de los modelos desarrollados y sus especificaciones?
<b>h) Pasos en el proceso de desarrollo</b>	¿Cuáles son los pasos? (Ej., Las tareas y las actividades) ¿La metodología especifica el proceso de desarrollo?
<b>i) Modelos y componentes notacionales</b>	¿Qué modelos (y/o componentes notacionales) son generados de cada paso?
<b>j) Definición de entradas y productos (salidas)</b>	¿Están definidos las entradas y los productos (salidas) en cada paso del proceso?
<b>k) Facilidad de conocimiento del proceso de cada paso</b>	¿Los pasos del proceso son fáciles comprender?
<b>l) Utilización de la metodología</b>	¿Los pasos del proceso son fáciles seguir?
<b>m) Refinabilidad (Finura):</b>	¿Los pasos de proceso proveen una ruta clara para mejorar los modelos de la metodología a través de etapas graduales para llegar a una puesta en práctica o, por lo menos, para conectar el nivel puesto en práctica a la especificación de diseño evidentemente?
<b>n) Enfoque dirigido al desarrollo de SMA</b>	<p>¿Cual es la metodología que?:</p> <p><b>a. ¿Enfoque de desarrollo de SMA es genérico?</b> (Ej., El enfoque es Orientado a Objetos o al enfoque de ingeniería de conocimientos)</p> <p><b>b. ¿Enfoque dirigido al uso de "Roles" en el desarrollo de SMA?</b> ¿La metodología emplea el concepto del "Rol (Papel)" en el análisis de SMA y diseño? Si es así, cual es el enfoque para la identificación de los roles (Ej., enfoque orientado a metas o objetivos, orientado al comportamiento, enfoque orientado a la organización.)</p>

Tabla 22. Documento 22 Criterios Relacionados Al Proceso  
Fuente: [HEGI05]

### 4.1.3. CRITERIOS RELACIONADOS A LA TECNICA

Los criterios relacionados a la técnica imponen la disponibilidad y la utilización de las técnicas suministradas de la metodología por la vía ejecutiva de los pasos del proceso y/o para el desarrollo de sus modelos y componentes de notacionales.

Documento 23 (Clave:Doc23)		Criterios Relacionados a La Técnica
a) Disponibilidad de las técnicas y las heurísticas	a) ¿Qué técnica usa para llevar a cabo cada paso del proceso? b) ¿Cuáles son las técnicas usadas para producir cada modelo y el componente notacional (Ej., Las técnicas de modelado)?	
b) Facilidad del conocimiento de la técnica	¿La técnica es fácil comprender?	
c) Usabilidad de la técnica	¿La técnica es fácil de seguir?	
d) Previsión de ejemplos	¿Los ejemplos de la técnica son fáciles de conseguir?	

Tabla 23. Documento 23 (Criterios Relacionados A La Técnica)  
Fuente: [HEGI05]

### 4.1.4. CRITERIOS RELACIONADOS AL MODELO

Los criterios relacionados al modelo valoran los varios aspectos de los modelos y componentes de notacionales de una metodología, incluyendo los conceptos representados por los modelos, la calidad de componentes notacionales, y el soporte del modelo de las características del agente.

Documento 24 (Clave:Doc24)		Criterios Relacionados Al Modelo
a) Conceptos	¿Qué conceptos pueden expresar o captar los modelos de la metodología?	
b) Completitud y expresividad	¿Los modelos de la metodología pueden representar al sistema de diferentes perspectivas, captando todos aspectos necesarios como los aspectos estáticos y dinámicos?	
c) Formalización / precisión de componentes notacionales	a) ¿La sintaxis y semántica de los componentes notacionales están definidos claramente? b) ¿Los ejemplos de los componentes notacionales están presentados?	
d) Derivación de modelo	¿Existen el proceso explícito y las pautas para transformar modelos en otros modelos o para crear un modelo de la información presente en otro modelo parcialmente?	
e) Consistencia	a) ¿Hay reglas y pautas para asegurar la consistencia entre los niveles de los conceptos abstractos dentro de cada modelo (Ej., La consistencia interna) y entre modelos diferentes?	

	b) ¿Los modelos son representados en una manera Ciclo de vida permite la regularidad de comprobación entre ellos?
<b>f) Complejidad:</b>	a) ¿Hay un número manejable de los conceptos expresados en cada modelo o componente notacionales? b) ¿Los componentes notacionales y la anotación son semánticamente y sintácticamente sencillos?
<b>g) Facilidad del conocimiento de modelos</b>	¿Los modelos y los componentes de notacionales son claros y fácil comprender?
<b>h) Modularidad</b>	¿La metodología y sus modelos promocionan la modularidad en el diseño de Agentes y el sistema?
<b>i) Abstracción</b>	¿La metodología tiene en cuenta crear modelos en varios niveles de detalle y/o abstracción?
<b>j) Autonomía</b>	¿Pueden los modelos soportar y representar las características autónomas de los agentes (Ej., La capacidad actuar sin la intervención directa de seres humanos u otros, y controlar sus propios estados y comportamiento)?
<b>k) Adaptabilidad</b>	¿Pueden los modelos soportar y representar el característica de capacidad de adaptación de agentes (Ej., La capacidad de aprender y mejorar con la experiencia)?
<b>l) Comportamiento cooperativo</b>	¿Pueden los modelos soportar y representar el comportamiento cooperativo de agentes (Ej., La capacidad de trabajar con otros agentes para conseguir un objetivo común)?
<b>m) Capacidad de deducción</b>	¿Los modelos pueden respaldar y representar la característica de la capacidad de deducir de los agentes (Ej., La capacidad de razonar y actuar en base a las especificaciones de operación)?
<b>n) Habilidad de comunicación</b>	¿Los modelos pueden respaldar y representar la capacidad de comunicación de "nivel de conocimientos" (Ej., La habilidad de comunicarse con otros agentes con un lenguaje parecido a los actos de habla del ser humano)?
<b>o) Personalidad</b>	¿Pueden los modelos soportar y representar la personalidad de agentes (Ej., La capacidad de manifestar los atributos de un personaje humano)?
<b>p) Reactividad</b>	¿Pueden los modelos soportar y representar la reactividad de agentes (Ej., La habilidad selectiva de intuir e interpretar en el momento oportuno)?
<b>q) Continuidad temporal</b>	¿Pueden los modelos soportar y representar la continuidad temporal de agentes (Ej. La perseverancia de la identidad y el estado sobre los períodos de tiempo largos)?
<b>r) Comportamiento deliberante</b>	¿Pueden los modelos soportar y representar el comportamiento deliberante de agentes (Ej., La capacidad de decidir pros y contras antes de decidir las)?
<b>s) Coincidencia</b>	¿Pueden los modelos soporte y representar la coincidencia en el procesamiento del agente (Ej., El logro de metas simultáneas o participación en las interacciones simultáneas)?

<b>t) Interacción con computadora humana</b>	¿Los modelos representan usuarios humanos y la interfaz de usuario?
<b>u) Reusabilidad de modelos</b>	¿La metodología provee o hacer él posible al uso de una biblioteca de modelos reutilizables?

**Tabla 24. Documento 24 (Criterios Relacionados Al Modelo)**  
Fuente: [HEGI05]

#### **4.1.5. CRITERIOS CARACTERISTICA ALENTADORA**

Los criterios alentador - característica tasan las características de alto nivel y adicionales varias de una metodología de AOSE. Las características consideradas incluyen la disponibilidad de herramientas de software y la soporte para ontología Agentes movible, y sistema abiertos.

<b>Documento 25 (Clave:Doc25)</b>	<b>Características Alentadora</b>
<b>a) Software y soporte metodológico</b>	¿La metodología es soportada por herramientas y bibliotecas?
<b>b) Sistemas abiertos</b>	¿La metodología provee el soporte para sistema abiertos (Ej., Sistema abiertos permiten la dinámica adición y corte de agentes)?
<b>c) Estructura dinámica</b>	¿La metodología suministra el soporte para la estructura dinámica (Ej., La metodología permite una dinámica modificación de la configuración del sistema)?
<b>d) La agilidad y la robustez</b>	¿La metodología provee el soporte para la agilidad y la robustez (por ejemplo., La metodología captura el procesamiento normal y el procesamiento de excepción, suministra las técnicas de analizar el rendimiento de sistema para todas configuraciones, o suministra las técnicas de detectar y recuperarse de fracasos)?
<b>e) Soporte para objetos convencionales</b>	¿La metodología provee el soporte para el uso o integración de objetos convencionales en un SMA (Ej., La metodología se dirige cómo pueden ser usados en uno SMA los objetos y cómo agentes Puede comunicarse con estos objetos)?
<b>f) Soporte para agentes móviles</b>	¿La metodología provee el soporte para el uso y la integración de agentes móviles en un SMA (Ej., La metodología hace un modelo de cuándo y cómo un agente debe ser móvil)?
<b>g) Soporte para ontología</b>	¿La metodología provee el soporte para el uso y la especificación de ontología de un SMA?

**Tabla 25. Documento 25 (Característica Alentadora)**  
Fuente: [HEGI05]

## 4.2. ANALISIS COMPARATIVO

### 4.2.1. COMPARACION DE LOS CRITERIOS RELACIONADOS AL PROCESO

Con respecto al enfoque de desarrollo de SMA genérico, la mayoría de las metodologías siguen el enfoque basado orientado a objeto (denotado como "OO") esto es; tomar la inspiración del paradigma de OO. También existe una metodología que asume el enfoque basado en la ingeniería de conocimientos (denotado como "KE" de *Knowledge engineering* la traducción al español es Ingeniería de conocimientos).

La comparación de criterios relacionados al proceso también trata el enfoque dirigido al uso de "Roles" en el desarrollo de SMA, se clasifica una metodología de AOSE como *non-role-oriented* ("NRO") en inglés, cuando no involucra el uso del concepto "Rol", por otro lado, si emplea el "Rol" como un concepto de modelado muy importante, se denota como *role-oriented* ("RO").

	Mas - COMMONAKADS	SMA de INGENIAS
<b>Desarrollo de Ciclo de vida (Doc22, a)</b>	Espiral dirigido por riesgos (Doc1, a)	Proceso Unificado de Desarrollo de Software (Doc4, a)
<b>La cobertura del Ciclo de vida (Doc22, b)</b>	Conceptualización, Análisis y Diseño (Doc2), (Doc8), (Doc9), (Doc10), (Doc11), (Doc12), (Doc13), (Doc14)	Análisis y diseño e Implementación (Doc15), (Doc16), (Doc17), (Doc18), (Doc19), (Doc20), (Doc21)
<b>Perspectiva de desarrollo (Doc22, c)</b>	Híbrido (Top-Down y Bottom-up) (Doc1, b)	Híbrido (Top-Down y Bottom-up) (Doc4, b)
<b>Dominio de aplicación (Doc22, d)</b>	Independiente (Doc1, c)	Independiente (Doc4, c)
<b>Tamaño de SMA (Doc22, e)</b>	No es especificado (Doc1, d)	No es especificado (Doc4, d)
<b>Naturaleza del Agente (Doc22)</b>	Heterogéneo e Híbridos (Doc1, e)	Híbridos (Doc4, e)
<b>Soporte para comprobación y validación (Doc22)</b>	Es mencionado pero no es explícito en pasos (Doc1, f)	Si (Doc4, f)
<b>Pasos en el proceso de desarrollo (Doc22)</b>	Si (Doc8), (Doc9), (Doc10), (Doc11), (Doc12), (Doc13), (Doc14)	Si (Doc15), (Doc16), (Doc17), (Doc18), (Doc19), (Doc20), (Doc21)
<b>Modelos y componentes notacionales (Doc22)</b>	Si (Doc3)	Si (Doc5), (Doc6)
<b>Definición de entradas y productos o salidas (Doc22)</b>	Si (Doc1), (Doc8, Notación de Casos de Uso), (Doc10, Descripción de Tareas), (Doc11, Descripción de conversaciones)	Si (Doc4), (Doc18), (Doc20), (Doc20)
<b>Facilidad de conocimiento del proceso de cada paso (Doc22)</b>	High (Doc1), (Doc8), (Doc9), (Doc10), (Doc11), (Doc12), (Doc13), (Doc14)	High (Doc4), (Doc15), (Doc16), (Doc17), (Doc18), (Doc19), (Doc20), (Doc21)
<b>Usabilidad de la metodología (Doc22)</b>	Médium (Doc1), (Doc8), (Doc9), (Doc10), (Doc11), (Doc12), (Doc13), (Doc14)	High (Doc4), (Doc15), (Doc16), (Doc17), (Doc18), (Doc19), (Doc20), (Doc21)
<b>Refinabilidad-Finura (Doc22)</b>	Si (Doc8), (Doc9), (Doc10), (Doc11), (Doc12), (Doc13),	Si (Doc15), (Doc16), (Doc17), (Doc18), (Doc19),

	(Doc14)	(Doc20), (Doc21)
<b>Enfoque dirigido al desarrollo de SMA (Doc22)</b>	a. KE (Doc1, g) b. RO(Doc9)	a. OO (Doc4) b. RO (Doc15)

**Tabla 26.** Comparación respecto a criterios relacionados al proceso

Algunas características son calificadas como “Alto” denotado en ingles High, “Medio” denotado como “Medium” y el nivel más “Bajo” de notado como “Low”.

#### **4.2.2. COMPARACION DE CRITERIOS RELACIONADOS A LA TECNICA**

Las metodologías son comparadas de acuerdo con los cuatro criterios relacionados con la técnica (presentado en la sección del mismo nombre) y los tres criterios que faltan relacionados al proceso (“*Pasos en el desarrollo del proceso*”, “*Modelos y componentes Notacionales*”, y “*Definición de Entradas y Salidas*”). La evaluación de técnicas es conectada con la valoración de cada paso y cada modelo por que las técnicas son dadas juntos con los pasos y los modelos.

Si una metodología es hallada como para soportar un paso especial, este soporte es denotado como (“E”) explícito o (“I”) Implícito. Si una metodología de AOSE contiene un paso importante en el desarrollo de su ciclo de vida y no es incluido en la lista, este paso será añadido como un “Extra” al final de la lista. Todas metodologías alcanzan a especificar productos o salidas para soportar sus pasos (denotado como “O” en las columna “Entradas / productos (salidas)”), mientras que si la mayoría de las metodologías especifican las entradas y salidas (denotado como “B” por Both en español Ambos).



Mas- CommonKADS								
Pasos	¿Soporte?	¿Modelos / Componentes Notacionales?	¿Entradas / salidas (productos)?	Técnicas por paso	Técnica para el modelado	Facilidad de entendimiento	Usabilidad	Ejemplos
1 Identifica los objetivos del sistema	E	Casos de Uso (Doc8)	B	Fase de Conceptualización (Doc8)	Describe metas y objetivos generales, también obtiene requerimientos en cuanto al Sistema (Doc8)	H	H	Y
2 Identifica las tareas y el comportamiento del sistema	E	Modelo de Tareas (Doc10)	O	Con un enfoque Top-down descompone las tareas (Doc1), (Doc10)	Se describen en forma de árbol (Doc10) La descripción de una tarea incluye su nombre, el tipo, condiciones de activación y de finalización, parámetros de entrada y salida, y una descripción corta (Doc10)	H	H	N
3 Especificación de casos de usos	E	Casos de uso (Doc8)	B	En la Fase de Conceptualización se obtiene una descripción preliminar del problema Para esto utilizan casos de uso (Doc8)	Se analizan los actores de los casos de uso de finidos en la conceptualización (Doc8)	H	H	Y
4 Identifica roles								
5 Identifica clases de agente	E	Modelo de Agente (Doc9)	B	Se desarrollan y analizan instancias iniciales del modelo de agentes para identificar y describir a los agentes (Doc9)	Para identificar agentes se pueden emplear distintas estrategias como analizar los actores de los casos de uso de la fase de conceptualización (Doc8) y roles (Doc9) similares puede ser agrupados como un solo agente para simplificar la comunicación	H	M	Y
6 Conceptualización del dominio de modelo		Modelo de Experiencia (Doc12)	O		La técnica es para modelar las capacidades de razonamiento del agente para realizar sus tareas y alcanzar sus metas (Doc12)			
7 Especifica conocimiento entre clases de agente	E	Modelo de Coordinación (Doc11)	B	Describe las conversaciones entre agentes, sus interacciones (Doc11)	El agente puede ofertar la realización de determinadas tareas (Doc10), denominadas servicios, a otros agentes	H	M	Y
8 Define protocolos de interacción	E	Modelo de Coordinación (Doc11)	B	La coordinación puede ser con comunicación o sin comunicación (en este último caso, las intenciones de los agentes se infieren a partir de la información obtenida escrita en la puzarna de ofertas en el modelo de agente) (Doc11)	El modelo de coordinación sólo modela la coordinación realizada con comunicación (Doc11)	H	H	Y
9 Define el contenido de mensajes cambiados	E	Modelo de Coordinación (Doc11)	B	Atendiendo al número de participantes, puede distinguir entre conversaciones bilaterales, entre dos agentes, y conversaciones multilaterales, entre más de dos agente (Doc11, FIGURA 3.2)	Cuadro De Descripción de conversación (Tipo, Objetivo, Agente, Iniciador, Servicio, Precondición, Poscondición, Condición-terminación, tiempo de ejecución, Descripción) (Doc11, Cuadros 10, 11,12)	H	H	Y
10 Especifica la arquitectura del Agente	E	Modelo de Diseño (Doc14)	B	(Diseñado por la plataforma de desarrollo)		H	L	N
11 Define estados mentales del Agente (E), Objetivos, las creencias, los planes, los compromisos )	E	Modelo de Experiencia (Doc12)	B	Mecanismo de Razonamiento estrictamente matemático (Doc12)	Descripción de cuadro de Experiencia (representación, tipo, grado de confiabilidad, esquema de procesamiento) (Doc12, Cuadro 14)	H	M	Y
12 Define una interfaz de la conducta del Agente (E), Les capacidad, los servicios, los contratos )	I	Modelo de Organización (Doc13), y de Agente (Doc9)	O	Describir la organización de los agentes software y su relación con el entorno (Doc9)	Modeladas con tarjetas CRC (Doc9)	H	L	Y
13 Especifica la arquitectura de sistema	E	Modelo de Organización (Doc13)	B		Analizar desde una perspectiva de grupo las relaciones entre los agentes (Doc13)	H	M	Y
14 Especifica la Estructura organizacional/régimen de control/relaciones inter-agente	I	Modelo de Organización (Doc13)			No se aplica al objeto de estudio en este caso	H	L	Y
15 Ambiente del SMA (E) Recursos, instalaciones, las características)	E	Modelo de Diseño (Doc14)	B	Esta centralizado en la fase de Conceptualización (Doc8) y Plataforma de Desarrollo	Transformar las especificaciones del resto de modelos para que se puedan describir con un lenguaje de programación	H	M	N
16 Especifica mecanismo de interacción de agente y su ambiente								
17 Especifica herencia y agregación del agente		No se aplica al objeto de estudio en este caso		No se aplica al objeto de estudio en este caso				
18 Instancia las clases Agente	I	Modelo de Organización (Doc14)	O	No se aplica al objeto de estudio en este caso	No se aplica al objeto de estudio en este caso	H	L	N
19 Especifica la instancias de agentes y su despliegue								

Tabla 27. Evaluación de pasos y técnicas Mas-Commonkads

INGENIAS								
Pasos	¿Soporte?	¿Modelos / Componentes Notacionales?	¿Entradas / salidas (estructura)?	Técnicas por paso	Técnica para el modelado	Facilidad de entendimiento	Usabilidad	Ejemplos
1 Identifica los objetivos del sistema	E	Modelo de Objetivos y Tareas (Doc18)	B	Identifique de requisitos de sistema, objetivos (Doc18)		H	H	Y
2 Identifica las tareas y el comportamiento del sistema	E	Modelo de Objetivos y Tareas (Doc18)	B	Obtener las tareas de requisitos de sistema o de objetivos (Doc18)	Recoger las motivaciones del SMA, definir las acciones identificadas en los modelos de organización (Doc18)	H	H	Y
3 Especificación de casos de usos	E	Diagramas de caso de uso (Doc15)	0	Elaborar el problema (Doc15)	Técnica OO modelado de casos de Uso (Doc15)	H	H	Y
4 Identifica roles	I	Modelo de Agente (Doc17)	B	Identifica los roles del análisis de workflows (Doc17)	El agente juega un rol, que lo lleva a perseguir sus objetivos (Doc17)	H	M	Y
5 Identifica clases de agente	E	Modelo de Agente (Doc17)	B	Aplicar "Principio de racionalidad" existe en asociaciones entre el agente, sus objetivos y las tareas desempeñadas (Doc17)	Describir a cada agente en relación con sus roles, objetivos, estados mentales (Doc17)	H	H	Y
6 Conceptualización del dominio de modelo								
7 Especifica conocimiento entre agentes	E	Modelo de Interacción (Doc19)	B	En la fase de análisis (Doc15), identificar las interacciones importantes entre agentes. Se construye sobre agentes, roles, objetivos, interacciones y unidades de interacción (Doc19)	Participantes de función (agentes/roles) y objetivos ejercidos por cada interacción (Doc19)	H	H	Y
8 Define protocolos de interacción	E	Modelo de Interacción (Doc19)	B	En la fase de diseño, elaborado cada interacción con la descripción detallada de elementos cambiados (Ej. Los mensajes) (Doc19)	En las interacciones se ejecutan unidades de interacción (paso de mensajes, lectura y escritura en un espacio de tuplas) en las que hay un iniciador (emisor) y colaboradores (receptores) (Doc19)	H	H	Y
9 Define el contenido de mensajes intercambiados	E	Modelo de Interacción (Doc19)	B		Para cada nombre de mensaje, función de la operación, parámetros, guardiames y anotación de la secuencia (Doc19)	H	M	Y
10 Especifica la arquitectura del Agente								
11 Define estados mentales del Agente (E) Objetivos, las creencias, los planes, los compromisos)	E	Modelo de Agente (Doc17)	B	Determinar los "Estados mentales" que en este caso va dado por un fórmula matemática (Doc17)	El control del estado mental esta basado en objetivos y tareas (Doc18)	H	H	Y
12 Define una interfaz de la conducta del Agente (Ej., Las capacidad, los servicios, los contratos...)								
13 Especifica la arquitectura de sistema (E) , Visión general de todos componentes y sus conexiones)	I	Modelo de Organización (Doc20)	B	Definir flujos de trabajo, del estudio de estos flujos surgen nuevas interacciones que reflejan con detalle cómo se coordinan los participantes del flujo (Doc18)	En el modelo de organización se refleja los componentes del sistema y sus conexiones implícitamente (Doc18)	H	H	Y
14 Especifica la Estructura organizacional/tránsito de contribuciones inter-agente	E	Modelo de Organización (Doc18)	B	Incrementalmente estructura en relación con componentes de sistema (Ej., Agentes, los papeles, los recursos, y las aplicaciones) (Doc18)	Indicar cómo los componentes de sistema son agrupados, sus dependencias sociales interacciones, tareas, objetivos y roles (Doc18)	H	H	Y
15 Ambiente del SMA (E) Recursos, instalaciones, las características)	E	Modelo de Entorno (Doc21)	B	Identifique recursos y aplicaciones en el entorno incluyendo requisitos de sistema y agente requisitos (Doc21)	Recursos de modelo y aplicaciones como objetos (Doc21)	H	H	Y
16 Especifica mecanismo de interacción de agente y su ambiente	E	Modelo de Entorno (Doc21)	B	Definir los recursos con los que trabajará el SMA, y las aplicaciones o sistemas con los que interactuará. (Doc21)	Identificar restricciones en el comportamiento de los agentes mediante relaciones (Doc21)	H	H	Y
17 Especifica herencia y agregación del agente								
18 Instanciar las clases Agente								
19 Especifica la instancias de agentes y su despliegue								
Extra la puesta en práctica (implementación)	E	INGENIAS IDR		Trazar un mapa de los componentes de diseño de sistema y componentes concretos en el marco de desarrollo		H	H	Y

Tabla 28. Evaluación de pasos y técnicas INGENIAS

La facilidad del conocimiento y la usabilidad de los pasos son valorados como "Alto" denotado como "H" por el ingles High, "Medio" como "M", o el nivel más "Bajo" de notado como "L" por el ingles Low.

• **Comparación respecto a pasos y usabilidad de las técnicas**

Pasos	Mas - COMMONKADS	INGENIAS
1. Identificación de objetivos de sistema	H (Doc8)	H (Doc15), (Doc18)
2. Identificación de tareas / comportamiento del sistema	H (Doc9), (Doc10)	H (Doc15), (Doc18)
3. Especificación de casos de uso	H (Doc8)	H (Doc15)
4. Identifica los Roles	M (Doc9)	M (Doc17)
5. Identifica clases de agente	M (Doc9)	H (Doc17)
6. Conceptualización del modelo de dominio	H (Doc8)	
7. Especificidad de conocimiento entre agentes	H (Doc9)	H
8. Define protocolos de interacción	H (Doc11)	H (Doc19)
9. Define el contenido de intercambio de mensajes	H (Doc11)	M (Doc19)
10. Especifica la arquitectura del Agente	L (Doc11)	
11 Define estados mentales del agente. (Ej., Objetivos, creencias, planes, compromisos, etc.)	M (Doc12)	H(Doc17)
12. Define interfaz conductual del Agente de la (Ej., capacidad, servicios, los contratos, etc.)	L (Doc9), (Doc10)	M(Doc20)
13. Especifica la arquitectura de sistema (Ej., Visión general de todos componentes y sus conexiones)	M (Doc11)	H (Doc19) (Doc20)
14. Especifica Estructura organizacional /régimen de control / relaciones inter-agentes	L (Doc13)	H (Doc20)
15. Ambiente del Modelo de SMA (Ej., Recursos, instalaciones, características)	M (Doc9)	H (Doc21)
16. Especifica mecanismo de interacción del agente en el ambiente.		H (Doc21)
18. Instanciación clases de Agente	L	

Tabla 29. Comparación de pasos y usabilidad de la técnica

**4.2.3. COMPARACION DE CRITERIOS RELACIONADOS AL MODELO**

La aportación adicional del análisis comparativo de los criterios relacionados al modelo son los criterios "Los conceptos" es *presentada en el* Tabla 30, mientras que el es resumido en el Tabla 31.

Conceptos	Mas - COMMONKADS	INGENIAS
Objetivo del sistema	Objetivos y Casos de Uso (Doc8)	Modelo de Objetivos Y Tareas (Doc18)
Sistema de tarea / comportamiento	Modelo de Tareas (Doc10)	Modelo de Objetivos Y Tareas (Doc18)
Escenarios de casos de uso	Conceptualización (Doc8)	Diagrama de Caso de Usos (Doc15)
Roles	Modelo de Agente(Doc9)	Modelo de Organización (Doc20) E Interacción (Doc19)
Conceptualización de Dominio	Fase de Conceptualización (Doc9) y Modelo Experiencia (Doc12)	Casos de Uso
Tareas y objetivos del Agente	Modelo de Agente(Doc9)	Modelo de Agente (Doc17)
Asignación de roles al agente	Modelo de Agente (Doc9)	Modelo de Agente (Doc17)
Creencia/ conocimientos del Agente	Modelo Experiencia (Doc12)	Estados Mentales pero no en un modelo
Habilidad de servicio del Agente	Modelo de Agente (Doc9)	

	Modelo de Organización (Doc13)	
Plan /regla de razonamiento / método de solución de problemas del Agente	Modelo Experiencia (Doc12)	Modelo de Agente (Doc17)
Percepción de eventos del Agente	Modelo de Coordinación (Doc11)	
Arquitectura del Agente	Modelo de Diseño	
Conocimiento del Agente	Modelo de Coordinación (Doc11)	Modelo de Interacción (Doc19)
Protocolo de interacción	Modelo de Coordinación (Doc11)	Modelo de Interacción (Doc19)
Satisface el intercambio de mensajes	Modelo de Coordinación (Doc11)	Modelo de Interacción (Doc19)
Contrato / compromiso inter-agente		
Arquitectura de Sistema	Modelo de Organización (Doc13)	Modelo de Organización (Doc20)
Relación social organizativa	Modelo de Organización (Doc13)	Modelo de Organización (Doc20)
Facilidad de recurso del entorno	Modelo de Organización (Doc13) Y Modelo de Diseño (Doc14)	Modelo de Entorno (Doc21)
Caracterización del ambiente		Modelo de Entorno (Doc21)
Relación de agregación del Agente	No se aplica al objeto de estudio en este caso	
Relación de herencia del Agente	No se aplica al objeto de estudio en este caso	
Particularización del Agente	No se aplica al objeto de estudio en este caso	

**Tabla 30.** Comparación de criterios relacionados al modelo

La evaluación del siguiente cuadro fue realizada en bases a la investigación realizada en [HEGI05] (Anexo A. 1), y en base a la experiencia de aplicación de ambas metodologías.

	Mas - COMMONKADS	INGENIAS
Complejidad / expresividad (Doc24, b)	S y H	S y H
Formalización / precisión (Doc24, c)	a. S y M b. S y M	a. H b. S
Derivación de modelo (Doc24,d)	S y L	S
Consistencia (Doc24,e)	a. N y L b. S y H	a. S y M b. S y H
Complejidad (Doc24,f)	a. S y M b. S y H	a. N b. N
Facilidad del conocimiento (Doc24,g)	S y H	S y M
Modularidad (Doc24,h)	S y M	S y H
Abstracción (Doc24,i)	S y H	S y H
Autonomía (Doc24,j)	S y M	S y M
Adaptabilidad (Doc24,k)	N	P y L
Comportamiento cooperativo (Doc24,l)	S y M	S y H
Capacidad de comunicación (Doc24, m)	S y M	S y H
Capacidad Inferencial (Doc24,n)	S y L	S y M
Reactividad (Doc24, o)	S y M	S y M
Comportamiento deliberante (Doc24, p)	S y H	S y H

Personalidad (Doc24, q)	N	N
Continuidad temporal (Doc24, r)	N	S y M
Concurrencia (Doc24, s)	N	N
Interacción con computadora humana (Doc24, t)	S y L	S y H
Reusabilidad de modelos (Doc24, u)	S y H	P

**Tabla 31.** Comparación de criterios relacionados al modelo

S: Si N: No P: Posiblemente H=High M=Médium L=Low

#### 4.2.4. COMPARACION DE CRITERIOS CARACTERISTICA ALENTADORA

Esta aportación adicional comparativa de las metodologías en relación con sus características de alto nivel y adicionales son resumidas en el siguiente cuadro:

	Mas - COMMONKADS	INGENIA S
Soporte metodológico y de software (Doc25)	N	S
Sistemas Abiertos(Doc25)	N	N
Estructura Dinámica (Doc25)	N	N
Agilidad y robustez (Doc25)	S	N
Soporte convencional a Objetos (Doc25)	N	S
Soporte a Agentes Móviles (Doc25)	S(*)	N
Soporte a Ontologías (Doc25)	S	N

**Tabla 32.** Comparación de criterios Característica Alentadora

S: Si N: No P: Posiblemente

(\*) Según Anexo A.2, tabla de comparación de la Característica Alentadora de la investigación sobre la comparación de 10 metodologías [HEGI05], MAS-CommonKADS no Soporta Agentes Móviles. Pero si nos aplicamos MAS-CommonKADS con la Técnica de UER [IGGA], Mas-CommonKADS puede soportar Agentes Móviles.

## CAPITULO V

### CONCLUSIONES

#### 5.1. INTRODUCCION

El desarrollo de Sistemas Multi-Agentes involucra una serie de procesos y artefactos que requieren una aplicación, definición y tratamiento apropiado. Para esto, muchos autores propusieron distintos métodos que van desde algoritmos simples hasta métodos complejos, desarrollando así más de veinte metodologías. Las metodologías estudiadas son las más representativas de diferentes líneas de investigación en agentes: MAS-CommonKADS del enfoque basado en la Ingeniería del Conocimiento, e INGENIAS con un enfoque orientado a objetos.

MAS-CommonKADS es una de las principales metodologías para utilizar la ingeniería de conocimiento estructurada en cuanto a Sistemas Multi-Agentes. Ha sido desarrollado y validado gradualmente por muchas compañías y universidades, según lo observado al realizar la investigación de esta metodología, y según muchos artículos publicados en la Web es actualmente el estándar europeo para el análisis del conocimiento y el desarrollo de conocimiento intensivo de sistemas basados en la Ingeniería del Conocimiento [HEGI05]. Por otro lado INGENIAS provee una bien definida colección y notación actividades para controlar el proceso, las tareas del análisis, el diseño, la verificación, en el desarrollo de Sistemas Multi-Agente. Y para respaldar las actividades de INGENIAS el equipo de desarrollo de INGENIAS creó una herramienta (IDK). Según [HEGI05] uno de los propósitos en la concepción de esta metodología es integrar los avances progresivos en la tecnología de agentes, aunque INGENIAS fue desarrollada siguiendo el Enfoque Orientada a Objetos.

A continuación se presentan las conclusiones más relevantes que fueron obtenidos tras realizar la investigación. Adicionalmente se presentan recomendaciones de trabajos futuros y las líneas de trabajo que se abren para ampliar el alcance de la investigación.

## 5.2 CONCLUSIONES GENERALES

Como conclusión relevante de este trabajo de investigación y siguiendo los objetivos que se plantearon se puede decir que:

- ♦ Al modelar un SMA de Licitaciones usando las Metodologías MAS-CommonKADS e INGENIAS para formular un comparativo de ambas en las fases de Análisis y Diseño. En base a la aplicación practica de ambas metodologías y al realizar la comparación referentes a los modelos, el proceso, los pasos y la característica alentadora de cada metodología, el proceso de desarrollo que propone INGENIAS es excesivo cuando se trata de desarrollos reducidos. A diferencia de MAS-CommonKADS que dota de procesos de desarrollo adaptados al tamaño del problema, INGENIAS da la impresión de dedicarse exclusivamente a desarrollos de gran tamaño. Como INGENIAS dispone de una cantidad ingente de entidades y relaciones. Su uso mediante la herramienta de soporte INGENIAS IDK se facilita, pero se sigue requiriendo que el desarrollador revise la documentación del grupo GRASIA para entender que hace cada entidad y cual es el propósito de cada relación. El principal inconveniente de MAS-CommonKADS es que el nivel de detalle alcanzado en la descripción no es realizable sin el apoyo de herramientas de soporte. Lo que propone MAS-CommonKADS, entre otras cosas, no es una notación sino una lista detallada de elementos y relaciones a identificar en el sistema. Un desarrollador puede seguir la lista y generar la documentación requerida de forma manual.
- ♦ La mejor manera de proporcionar recomendaciones para el desarrollo de agentes usando las metodologías MAS-commonKads e INGENIAS es haciendo notar que la aplicación de una u otra metodología para el desarrollo de Sistemas Multi-Agente depende mucho de la formación que tenga su usuario final. Esto quiere decir que son interpretadas en función de los intereses de quien las aplica. En la actualidad, no hay soluciones que fuercen un uso en concreto. MAS-CommonKADS como INGENIAS constituyen un ejemplo a seguir en lo que a metodologías se refiere. Son exhaustivas como pocas a la hora de detallar el sistema y además es consecuente con que el proceso de desarrollo en la mayoría de los casos es más

complejo que un conjunto de pasos. De hecho, las ideas de MAS-CommonKADS han estado presentes en casi todos los trabajos referentes a metodologías desde hace años.

- ♦ Implementar el SMA de Licitaciones en la plataforma JADE para crear el prototipo de la tesis, aplicando los modelos de las metodologías estudiadas, coopera a comprender que si capturamos bien los requerimientos del SMA en los diferentes modelos de cada metodología, se hace mas fácil llevar estos a la programación y crear un SMA real y funcional, para cumplir los requerimientos para lo cual fue modelado el SMA

Y siguiendo la hipótesis planteada y al tratar de encontrar o no una diferencia significativa existente entre INGENIAS y MAS-CommonKADS, se puede decir que:

“En conclusión general las metodologías son significativamente iguales, si se está acostumbrado a trabajar con sistemas basados en conocimiento como con la metodología CommonKADS, lo lógico es que se elija MAS-CommonKADS. Si por el contrario está interesado en un enfoque más orientado a Agentes o a Objetos puede elegir y además de contar con un soporte de desarrollo debería de elegir INGENIAS. En los casos en que se requiera un proceso de desarrollo robusto, detallado y ensayado en desarrollos reales, la recomendación sería MAS-CommonKADS o INGENIAS pues ambas reúnen estas características”.

### **5.3. RECOMENDACIONES**

- En el área de ingeniería de Software se recomienda crear técnicas de evaluación sujeto a normas vigentes para la construcción acertada de SMA debido a que a las técnicas de evaluación actuales no siguen un estándar.
- Aplicar muchas mas metodologías este caso de estudio u otro caso para obtener conclusiones que puedan ser generalizadas y aplicadas para la elección de una metodología apropiada a la del usuario.
- Proponer un estudio en cuanto a la fase de mantenimiento de todo SMA y la comparación que pudiera tener con métodos tradicionales de desarrollo, asociado a corrección de errores, debido a la evolución del entorno de software.



## REFERENCIA BIBLIOGRAFICAS

### Libros:

1. Roberto Hernández Sampieri, Carlos Fernández, Pilar Baptista, “*Metodología de la Investigación*” 3ra. Edición, Editorial McGraw Hill, 2003. 705 Pág. ISBN 970-10-3632-8
2. Roger S. Pressman, “*Ingeniería del Software*”, 5º Edición, Editorial Concepción Fernández Madrid, 2005.958 Pág. ISBN 970-10-5473-3
3. Guillermo Choque Aspiazu “*Inteligencia Artificial Perspectivas y Realizaciones*” 2002
4. Iliana Jaque, Ximena Sánchez, Héctor Gómez. Guía para la presentación de Referencias bibliográficas de publicaciones impresas y electrónicas [en línea]. Santiago de Chile: Universidad Tecnológica Metropolitana. Julio 2003 [Consultado: Junio-2008]. Disponible en : < <http://cv.umsa.bo/informatica/mod/resource/view.php?id=736> >
5. [HEGI05] Henderson-Sellers Brian (University of Technology, Sydney, Australia), and Giorgini, Paolo (Trento University, Italy). “Agent Oriented Methodologies”.Published in the United States of America by Idea Group Publishing (an imprint of Idea Group Inc.).2005. 429pag. Paginas consultadas: 347-367.ISBN 1-59140-581-5 (hardcover) ISBN 1-59140-586-6 (soft cover) -- ISBN 1-59140-587-4(ebook)

### Tesis y Proyectos de Grado

1. Richard Rubén Velasco Ali. “Estandarización de Metodologías para el Desarrollo de Sistemas Multi-Agente”.Tesis de Grado para optar al Titulo de Licenciatura en Informática mención en Ingeniería de Sistemas Informáticos. La Paz, Bolivia: Universidad Mayor de San Andrés. 2006. 200 Pág.

### Documentos:

[ARA05] ARANDA Corral, Gonzalo A. Curso de Programación de Agentes.Seminario de Jade Sevilla, 23 de Mayo de 2005,33 Diapositivas.

[BON88] A. H. Bond and L. Gasser. An analysis of problems and research in DAI. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 3–36. Morgan Kaufmann Publishers: San Mateo, CA, 1988.

[FMS97] M. Fisher, J. Müller, M. Schroeder, G. Staniford, and G. Wagne. Methodological foundations for agent-based systems. In Proceedings of the UK Special Interest Group on Foundations of Multi-Agent Systems (FOMAS). Published in Knowledge Engineering Review (12) 3, 1997,. <http://www.dcs.warwick.ac.uk/fomas/fomas96/abstracts/ker3.ps>.

[GAR05] Garamendi Bragado, Juan Fco. Agentes Inteligentes: JADE [en línea]. Universidad Rey Juan Carlos, Programa de Doctorado: Informática y Modelización Matemática Universidad Rey Juan Carlos. Abril 2004[Consulta: 31-marzo-2009]. Disponible en: < <http://www.agentintelligent.com.es> >

[GRA02] GRASIA Grupo de Agentes de Software: Ingeniería y Aplicaciones [en línea]. Madrid: Universidad Complutense de Madrid Facultad de Informática 2002. [Consulta: 25-junio- 2009]. Disponible en: < <http://grasia.fdi.ucm.es> >

**[IGGA]** IGLESIAS, Carlos A. GARIJO, Mercedes. UER Technique Conceptualisation for Agent Oriented Development. Universidad Politécnica de Madrid, España. Departamento Ingeniería de Sistemas Telemáticos,. Articuló publicado en formato PDF. [Consultado el 21 Mayo de 2009] disponible en <<http://www.universidadpolitecicamadrid.es/uertechnique>>

**[IGL98]** IGLESIAS Fernández, Carlos Ángel. Definición de una metodología Para el desarrollo de sistemas multiagente. Tesis Doctoral. Departamento De Ingeniería De Sistemas Telemáticos, Universidad Politécnica De Madrid, España, 1998. 332 Págs

**[JEJ95]** N. R. Jennings and A. J. Jackson. Agent-based meeting scheduling: A design and implementation. *Electronic Letters, The Institution of Electrical Engineering*, 31(5):350–352, March 1995.

**[JEW95]** N. R. Jennings and M. Wooldridge. Applying agent technology. *Applied Artificial Intelligence*, 9(6):357–370, 1995.

**[MAR00]** MARCHETTI, Tulio José; GARCIA, Alejandro Javier. Metodologías de desarrollo de sistemas multi-agente: un análisis comparativo, Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA) Departamento de Ciencias e Ingeniería de la Computación Universidad Nacional del Sur, 2000.121pags.

**[MUL96]** H. J. Müller. (Multi)- agent systems engineering. In Second Knowledge Engineering Forum, Karlsruhe, Feb. 1996.

**[PAR05]** PAREJAS, César. Análisis Crítico de la Situación actual de la tecnología de agentes en el desarrollo de sistemas Multi-Agentes para dispositivos móviles. Doctorado En Ingeniería Informática. UNIVERSIDAD PONTIFICIA DE SALAMANCA, Facultad de Informática Campus de Madrid, Febrero de 2005. 122pag

**[PAV04]** PAVON, Juan. Agentes Inteligentes Desarrollo de Sistemas Multi-Agente con INGENIAS, Curso de doctorado. Universidad Complutense de Madrid, Dep. de Sistemas Informáticos y Programación, España. Articuló publicado en diapositivas, 2004. Disponible en <<http://grasia.fdi.ucm.es>>

**[PAV05]** Pavón, Juan. Agentes inteligentes Comunicación entre agentes [en línea]. Madrid: Universidad Complutense de Madrid Facultad de Informática Dep. de Sistemas Informáticos y Programación 2003-2005. [Consulta: 18-mayo- 2009]. Disponible en: <<http://grasia.fdi.ucm.es>>

**[ROD07]** Dr. Rodríguez Elías Oscar Mario. Tópicos Avanzados de Ciencias de la Educación [en línea]. Diapositiva 1 al 12 Metodologías de desarrollo: MAS-CommonKADS. Universidad de Sonora, Departamento de Matemáticas, Licenciatura en Ciencias de la Computación.[Consulta: 31-marzo-2009]. Disponible en: <[www.mat.uson.mx/omrodriguez](http://www.mat.uson.mx/omrodriguez)>

**[WEBJADE]** Pagina oficial del proyecto JADE <<http://sharon.cselt.it/projects/jade/>>



# ANEXOS



# ANEXO A

---

## TABLAS DE COMPARACION DE [HEGI05]

**Comparación de criterios relacionado al Modelo**  
**(Comparison regarding model related criteria)**

CONCEPTOS	GAIA	TROPOS	MAS-COMMONKADS	PROMETHEUS	PASSI	ADELFE	MASE	RAP	MESSAGE	INGENIAS
Compleitud / expresividad	M	H	H	H	H	H	H	M	M	H
Formalización / precisión	a.H b.Y	a.H b.Y	a.M b.Y	a.H b.Y	a.H b.Y	a.H b.Y	a.H b.Y	a.H b.Y	a.H b.Y	a.H b.Y
Derivación de modelo	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Consistencia	a.Y b.Y	a.Y b.Y	a.N b.Y	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.N b.Y	a.N b.Y	a.Y b.Y
Complejidad	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.Y b.Y	a.N b.N
Facilidad e Conocimiento	H	H	H	H	H	H	H	H	H	M
Modularidad	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Abstracción	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
Autonomía	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Adaptabilidad	P	N	N	N	N	Y	N	N	P	P
Comportamiento Cooperativo	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Capacidad Comunicación	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Capacidad Inferencial	N	Y	Y	Y	Y	Y	P	P	Y	Y
Reactividad	P	Y	Y	Y	Y	Y	Y	Y	Y	Y
Comportamiento Deliberativo	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
Personalidad	N	N	N	N	N	N	N	N	N	N
Continuidad Temporal	N	N	N	N	N	N	N	N	N	Y
Concurrencia	N	N	N	N	Y	N	Y	N	N	N
Interacción con Computadora Humana	N	Y	Y	Y	Y	Y	N	Y	Y	Y
Reusabilidad de Modelos	Y	P	Y	P	Y	P	Y	P	P	P

Anexo A. 1. Tabla Comparación de criterios relacionado al Modelo

Fuente [Table 5, Comparison regarding model related criteria, Pag. 364, HEGI05]

## Comparación de Criterio relacionado a la Característica Alentadora

*(Comparison regarding supportive related criteria)*

CONCEPTOS	GAIA	TROPOS	MAS-COMMONKADS	PROMETHEUS	PASSI	ADELFE	MASE	RAP	MESSAGE	INGENIAS
Soporte metodológico y de software	N	N	N	S	S	S	S	S	S	S
Sistemas Abiertos	S	N	N	N	N	S	N	N	N	N
Estructura Dinámica	N	N	N	N	S	N	P	N	N	N
Agilidad y robustez	N	N	S	S	N	S	S	S	N	N
Soporte convencional a Objetos	N	N	N	S	N	S	N	S	S	S
Soporte a Agentes Móviles	N	N	N	N	S	N	N	N	N	N
Soporte a Ontologías	N	N	S	N	S	N	N	N	S	N

Anexo A.2. Tabla Comparación de criterios relacionado al Modelo

Fuente [Table 6, Comparison regarding supportive related criteria, Pag. 365, HEG105]



## ANEXO B

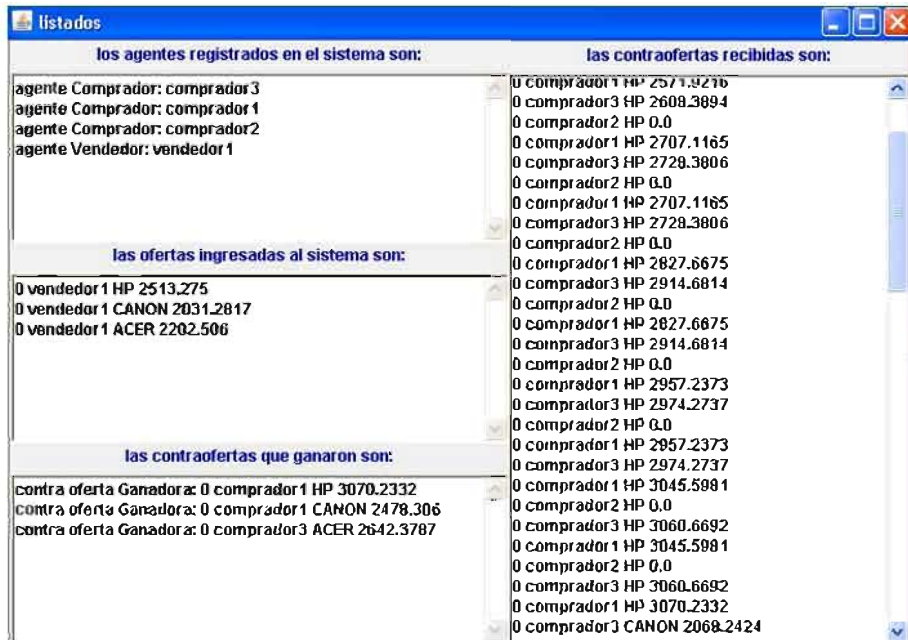
---

### PRUEBAS DE PROTOTIPO DE SMA DE LICITACIONES

## CASO # 1 (TRES COMPRADORES Y UN VENDEDOR CON 3 ARTICULOS)

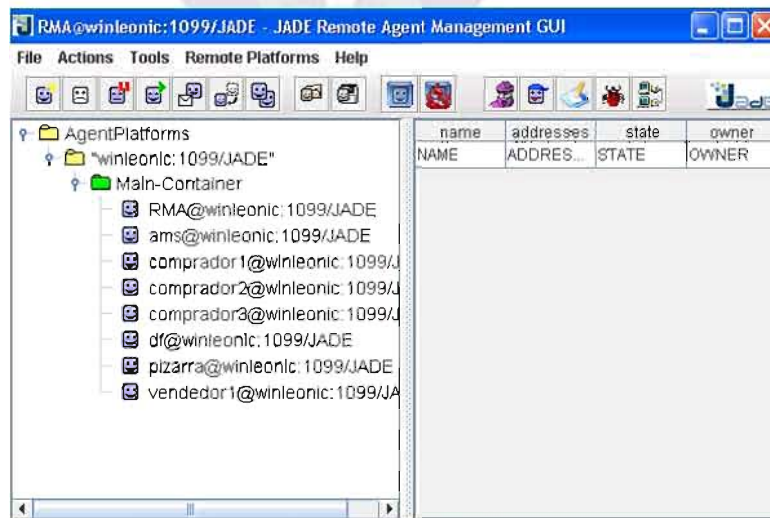
### PARAMETROS DE ENTRADA:

-gui pizarra:licitacion.Pizarra comprador1:licitacion.Comprador(4500.0)  
 comprador2:licitacion.Comprador(2000.0) comprador3:licitacion.Comprador(3550.0)  
 vendedor1:licitacion.Vendedor(HP 2500.0 CANON 2000.0 ACER 2200.0 )



**ANEXO B.1. Pizarra Licitación Caso # 1**  
 Fuente [Eclipse, Elaboración Propia]

Se observa a los agentes registrados en la pizarra ANEXO B.1 y en la plataforma JADE ANEXO B.2, tres compradores y un vendedor, artículos a la venta, contraofertas, y contraofertas ganadoras.



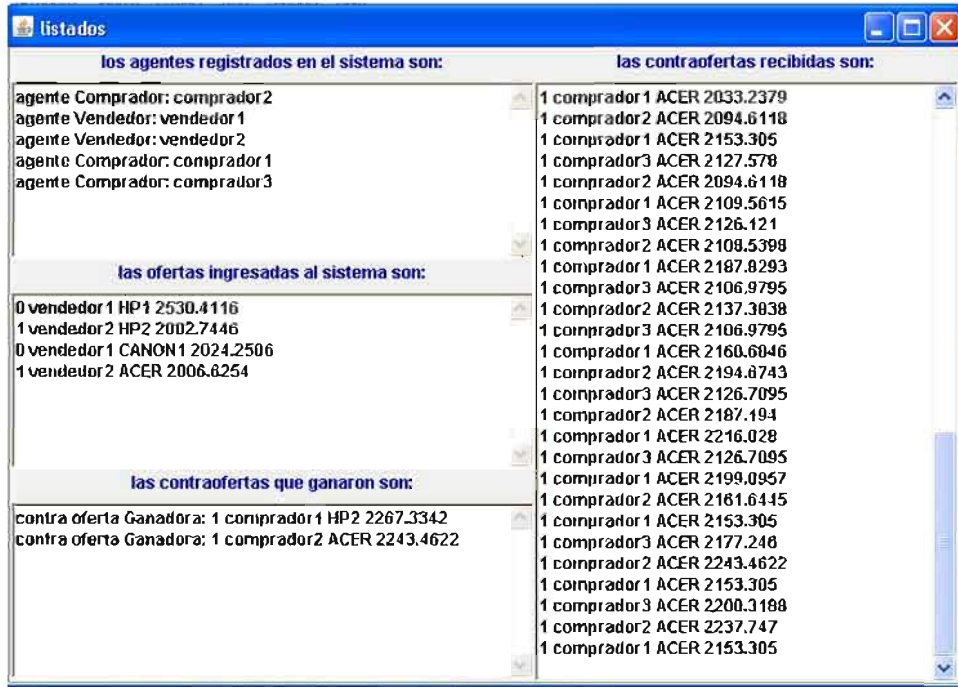
**ANEXO B.2. JADE Licitación Caso # 1**  
 Fuente [Eclipse, Elaboración Propia]



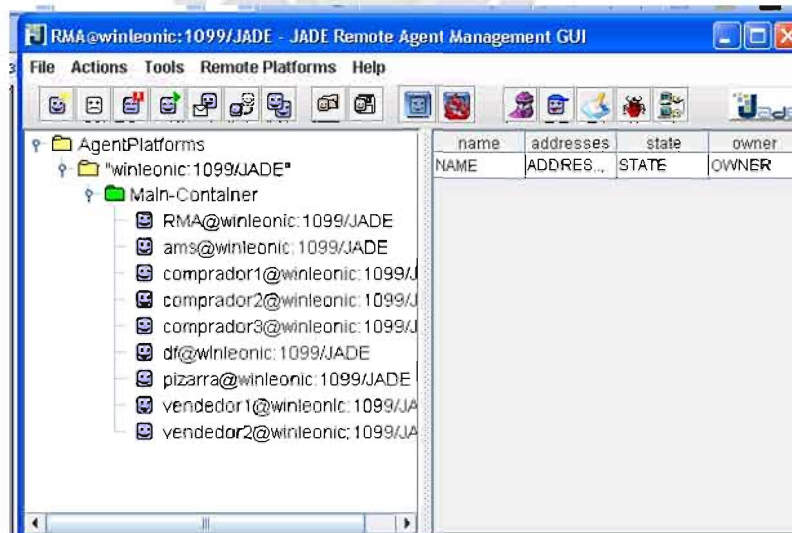
## CASO # 2 (TRES COMPRADORES Y DOS VENDEDORES CON 3 ARTICULOS)

### PARAMETROS DE ENTRADA:

-gui pizarra:licitacion.Pizarra comprador1:licitacion.Comprador(8500.25)  
 comprador2:licitacion.Comprador(6000.0) comprador3:licitacion.Comprador(4000.0)  
 vendedor1:licitacion.Vendedor(HP1 2500.0 CANON1 2000.0)  
 vendedor2:licitacion.Vendedor(HP2 2000.0 ACER 2200.0)



**ANEXO B.3. Pizarra Licitación Caso # 2**  
 Fuente [Eclipse, Elaboración Propia]

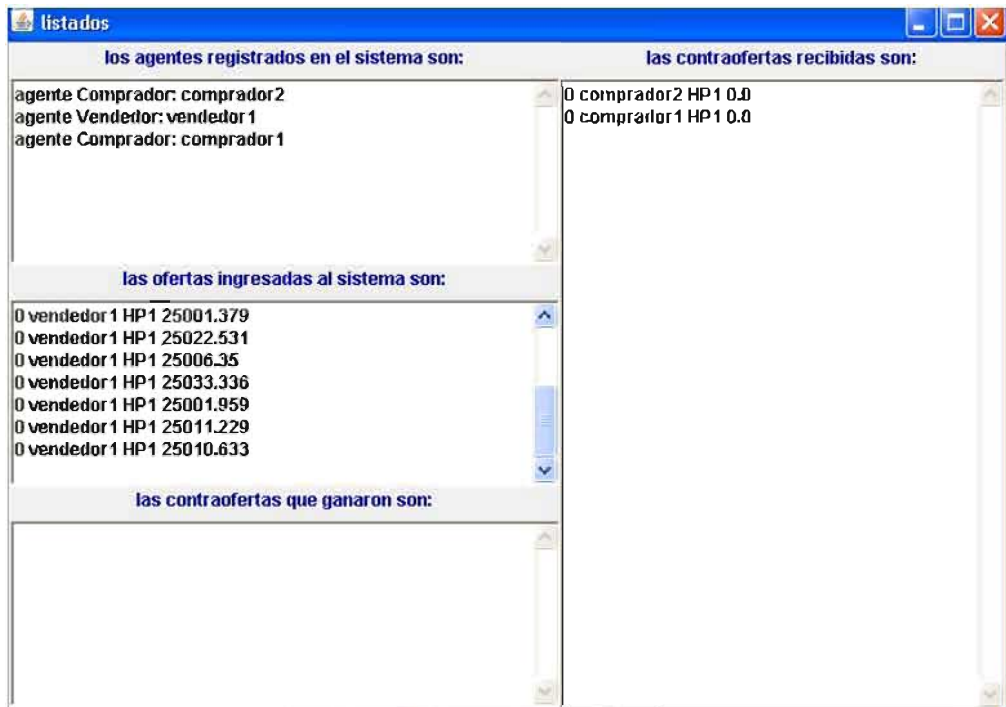


**ANEXO B.4. JADE Licitación Caso # 2**  
 Fuente [Eclipse, Elaboración Propia]

**CASO # 3 (DOS COMPRADORES Y UN VENDEDORES CON EL PRIMER ARTICULO QUE SOBREPASA LA CANTIDAD DE DINERO DE LOS COMPRADORES)**

PARAMETROS DE ENTRADA:

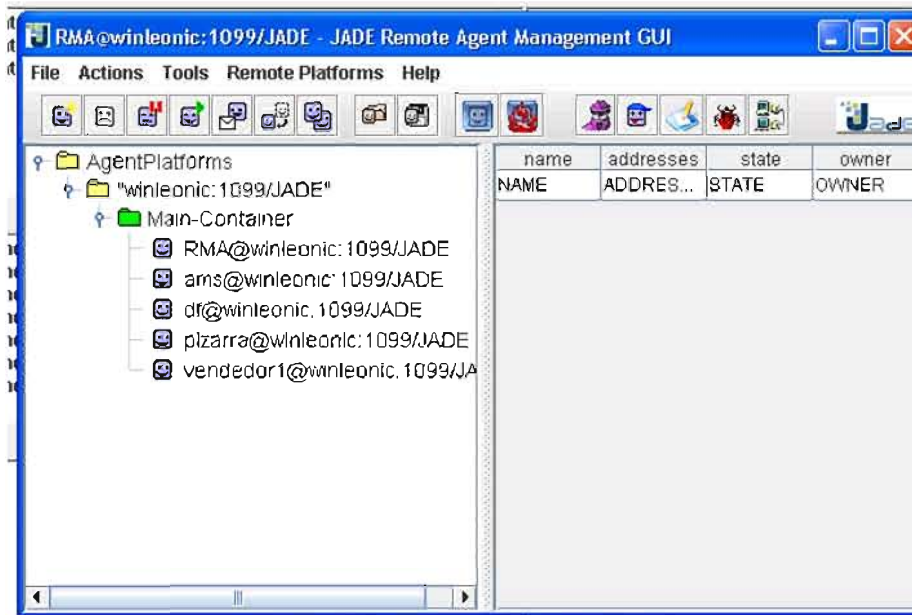
-gui pizarra:licitacion.Pizarra comprador1:licitacion.Comprador(4500.0)  
comprador2:licitacion.Comprador(2000.0) vendedor1:licitacion.Vendedor(HP1 25000.0  
CANON1 2000.0)



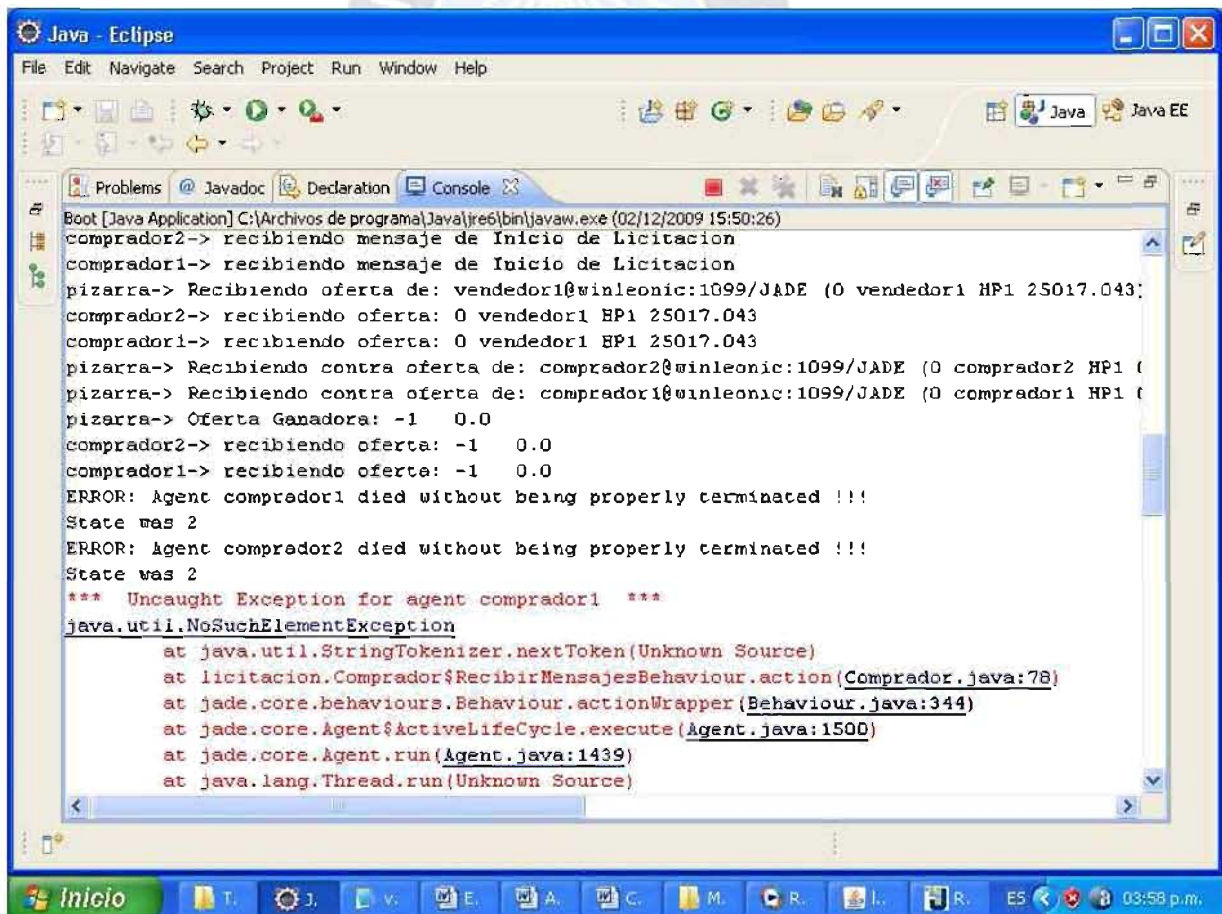
**ANEXO B.5. Pizarra Licitación Caso # 3**  
**Fuente [Eclipse, Elaboración Propia]**

Como observamos en la Pizarra ANEXO B.5 del Caso # 3 vemos que el sistema registra al vendedor y a los compradores. Pero como los compradores no cuentan con la cantidad suficiente como para alcanzar a ofertar por el primer articulo, el sistema se para liza y ya no continua. También podemos observar que en este caso como vemos en ANEXO B.6 JADE no reconoce a los compradores, pero si al vendedor y al agente pizarra.

Si observamos la consola de Eclipse para el Caso # 3 ANEXO B.7, vemos que el programa se ejecuta pero con errores, y por ende elimina a los agentes registrados



ANEXO B.6. JADE Licitación Caso # 3  
Fuente [Eclipse, Elaboración Propia]

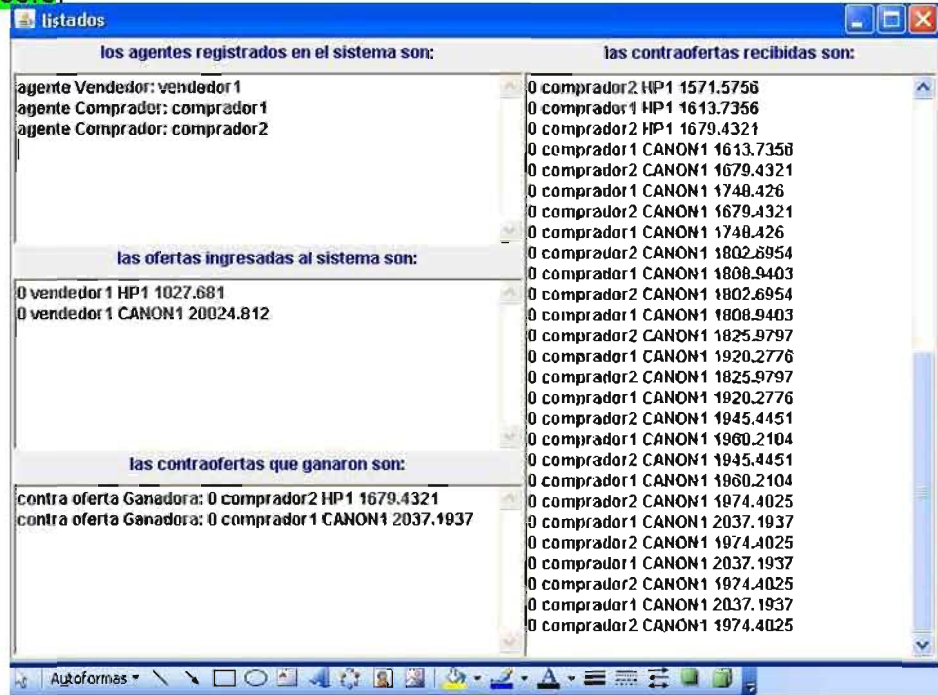


ANEXO B.7. Consola Eclipse para Licitación Caso # 3  
Fuente [Eclipse, Elaboración Propia]

**CASO # 4 (DOS COMPRADORES Y UN VENDEDORES CON EL SEGUNDO ARTICULO QUE SOBREPASA LA CANTIDAD DE DINERO DE LOS COMPRADORES)**

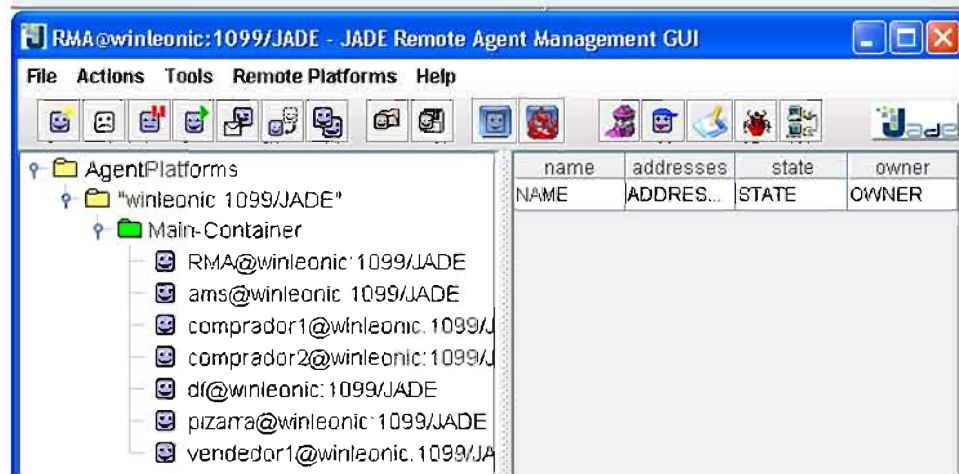
PARAMETROS DE ENTRADA:

-gui pizarra:licitacion.Pizarra comprador1:licitacion.Comprador(4500.0)  
 comprador2:licitacion.Comprador(2000.0) vendedor1:licitacion.Vendedor(HP1 1000.0  
 CANON1 2000.0)



**ANEXO B.8 Pizarra Licitación Caso # 4**  
 Fuente [Eclipse, Elaboración Propia]

Como se ve en ANEXO B.8 el artículo se vende a un precio muy inferior, debido a que los compradores no pueden ofertar más alto, y en este caso si son registrados todos los agentes como se ve en ANEXO B.9



**ANEXO B.9 JADE Licitación Caso # 3**  
 Fuente [Eclipse, Elaboración Propia]