



# Avances en la Inteligencia Artificial Distribuida

Jose Aguilar  
CEMISID, Escuela de Sistemas  
Facultad de Ingeniería  
Universidad de Los Andes  
Mérida, Venezuela



# Sistemas Multi-agentes

# Agente Inteligente

- Sabe plantearse un problema y buscarle soluciones  
=> técnicas de búsqueda y resolución de problemas
- Tiene formas de representar el conocimiento, razonar y manejar la incertidumbre  
=> Lógicas y Mecanismos de manejo de conocimiento, incluso incierto, para posibilitar las diferentes formas de razonamiento
- Es capaz de planificarse y aprender
- Es capaz de comunicarse  
=> **Sistemas Multiagentes**

# SMA

- Son sistemas compuestos por múltiples componentes *autónomos* que poseen las siguientes características:
  - Cada agente tiene ciertas capacidades.
  - Necesitan de formas de coordinarse
  - Los datos no están centralizados
  - La computación es asíncrona.

SMA	RPD
Se enfocan primordialmente en la coordinación o negociación entre los agentes	Se enfocan en la resolución de un pb. dado, su descomposición, y síntesis de la soluciones
Son autónomos con la posibilidad de objetivos individuales	Necesitan sincronizarse
Permite la emergencia	Es determinista



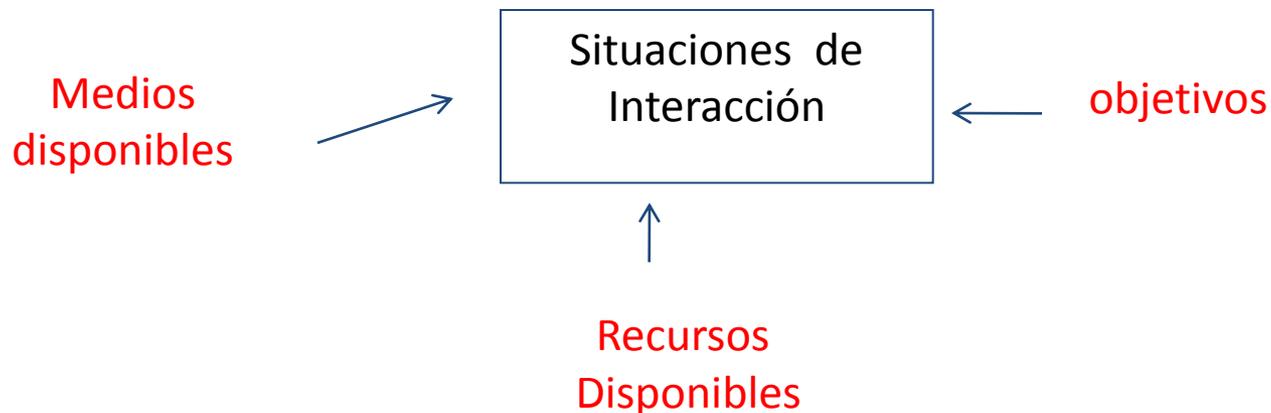
# Características de ambientes multiagentes

- Proveen una infraestructura específica de comunicación y protocolos de interacción.
- Son típicamente abiertos y no tienen un diseño centralizado.
- Contienen agentes que son autónomos y distribuidos, y pueden ser interesados en si mismos o cooperar.

# INTERACCIONES

# Interacciones

Una situación de **interacción** se produce cuando **dos o más agentes, por medio** de una **serie de acciones recíprocas** confluyen en una **relación dinámica**. Estas acciones, las influencias que unos ejercen sobre los otros, las obligaciones y compromisos que establecen, hacen de los agentes **entidades sociales**.



# Interacciones

- La interacción en un SMA está determinada por una serie de protocolos de comunicación que establecen los *actos de habla* válidos que permiten intercambiar información entre los agentes para lograr un objetivo.
- Entre los protocolos más utilizados se pueden nombrar el de Solicitud (Request), Consulta (Query), Redes de Contrato (Contract-Net), Subasta Inglesa (British Auction) y Subasta Holandesa (Dutch Auction).

# INTERACCIONES

OBJETIVOS	RECURSOS	CAPACIDADES	TIPO	CATEGORIA
COMPATIBLES	SUFICIENTES	SUFICIENTES	<i>INDEPEND.</i>	<b>INDIFERENTE</b> (PERS. CAMINA ACERA)
COMPATIBLES	SUFICIENTES	INSUFICIENTES	<i>COLABOR. SIMPLE</i>	<b>COOPERACION</b> (ASIGNACION DE TAREAS)
COMPATIBLES	INSUFICIENTES	SUFICIENTES	<i>ENTRABAR</i>	<b>COOPERACION</b> (CIRCULAC. CARROS, USO PROCESADOR, TRAF. AER)
COMPATIBLES	INSUFICIENTES	INSUFICIENTES	<i>COLABOR. COORD.</i>	<b>COOPERACION</b> (FABRICACION INDUSTRIAL)
INCOMPATIBLES	SUFICIENTES	SUFICIENTES	<i>COMPET. INDIV. PURA</i>	<b>ANTAGONISMO</b> (MARATON)
INCOMPATIBLES	SUFICIENTES	INSUFICIENTES	<i>COMPET. COLECT. PURA</i>	<b>ANTAGONISMO</b> (CARRERA DE RELEVO)
INCOMPATIBLES	INSUFICIENTES	SUFICIENTES	<i>CONFLICTO INDIV. RECUR.</i>	<b>ANTAGONISMO</b> (DEFENSA TERRITORIO, JEFE MANADA)
INCOMPATIBLES	INSUFICIENTES	INSUFICIENTES	<i>CONFLICT. COLECT RECUR.</i>	<b>ANTAGONISMO</b> (GUERRAS, COMPET. INDUSTRIAL)

# Tipos más comunes de interacción

- **Coordinación:** propiedad de interacción entre un conjunto de agentes que realizan alguna actividad colectiva:
  - **Coordinación global:** el sistema determina y planifica globalmente, existiendo un agente que verifica todos los conflictos.
  - **Coordinación individual:** cada agente tiene completa autonomía para decidir qué hacer y resolver los conflictos que detecte localmente.
- **Cooperación:** una clase de coordinación que consiste en que varios agentes no antagonistas (acciones de uno no perjudiquen al otro) interactúen entre sí con el objetivo de *conseguir un fin común*.
- **Negociación:** consiste en poner de acuerdo a los agentes de un sistema, cuando cada uno defiende sus propios intereses, llevándolos a una *situación que los beneficie a todos* (comercio electrónico, subastas, ...)

# Cooperación

# COOPERACION

*proceso por el que ciertos agentes participantes generan deberes mutuamente dependientes para actividades conjuntas (planes).*

Las situaciones de cooperación aparecen cuando los agentes tienen que resolver problemas o tareas interdependientes

- Se recibe un problema en cierto nivel de abstracción.
- El agente resuelve localmente aquello que es posible.
- Recurre a otros agentes del mismo nivel para el resto de las tareas.
- Recurre a otros niveles de abstracción para el resto de las tareas.

# COOPERACION

## RAZONES DE LA COOPERACION

- MEJORAR CAPACIDAD DE SOBREVIVIR
- INCREMENTAR RENDIMIENTOS
- RESOLVER CONFLICTOS

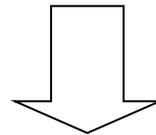
<b>VENTAJAS VS. AMPLIFICACION DE CAPACIDADES</b>	<b>CUALITATIVAS</b>	<b>CUANTITATIVAS</b>
<b>CUALITATIVAS</b>		<b>CATALISIS</b>
<b>CUANTITATIVAS</b>	<b>EFECTO DE MASAS</b>	

# COLABORACION Y REPARTICION DE TAREAS

Repartir tareas, recursos e información implica:

- Quién debe hacer qué con qué medios

En función de los objetivos y competencias de los agentes y restricciones del ambiente



Descomposición de Tareas

# COLABORACION Y REPARTICION DE TAREAS

- **ROLES**

CLIENTES O DEMANDADORES  
PROVEEDORES O SERVIDORES  
MEDIADORES

- **FORMAS DE ASIGNACION**

PREDEFINIDA

CENTRALIZADA

- JERARQUICO => IMPUESTA

- IGUALITARIO => MEDIADOR

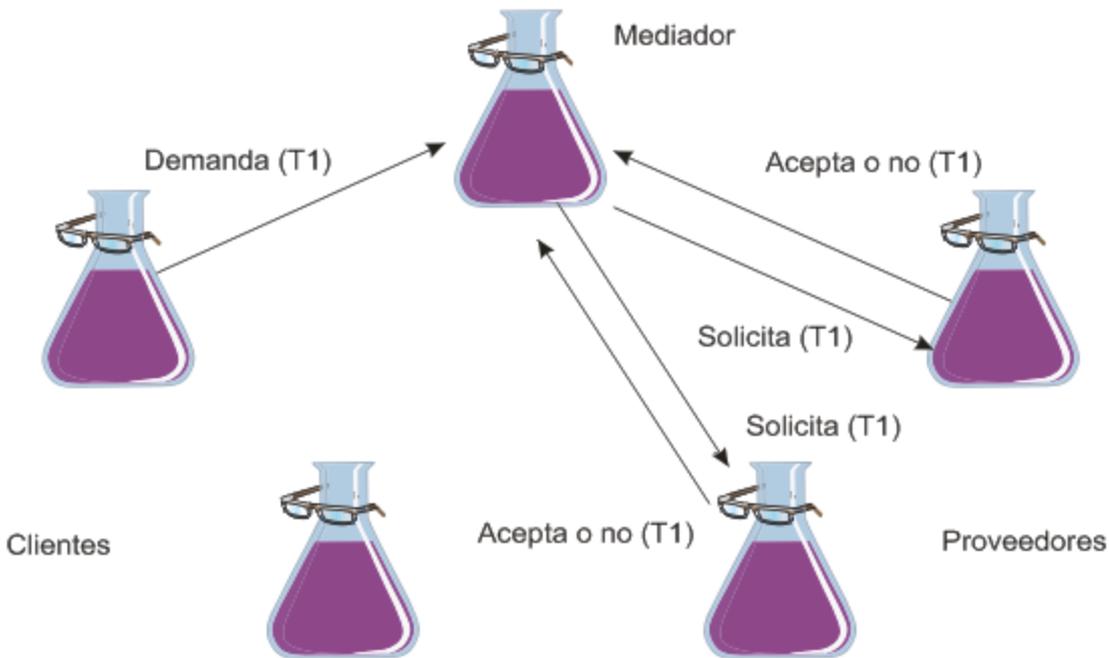
DISTRIBUIDA

- RED DE CONTACTOS: directa o delegar

- RED DE CONTRATO: licitación o subasta

EMERGENTE

# Asignación de Recursos



Centralizada clásica.

Centralizada con mediador.

Por redes de amistades.

por concurso

Emergente

# ASIGNACION EMERGENTE

- ASIGNACION REACTIVA
  - **SENALES**
  
- ARRANQUE DIFERENCIAL DE UN COMPORTAMIENTO EN FUNCION DE LA SENAL QUE RECIBE UN AGENTE
  - **COMPETENCIA INTRA-AGENTE:**
    - Intensidad de la señal
    - Tendencia a cumplir cierta tarea (ya comí?, etc.)
    - Distancia a la fuente y capacidad de percibir
  - **RELACION AGENTE VS. FUENTE DE LA SENAL**

## UMBRAL DE RESPUESTA

# Coordinación

# COORDINACION DE ACCIONES

*CONJUNTO DE ACTIVIDADES SUPLEMENTARIAS QUE SE DEBEN REALIZAR EN UN SMA*

Tareas no directamente productivas que mejoran la eficiencia total del sistema. Los agentes se coordinan para gestionar las dependencias existentes entre sus actividades

- ARTICULACION DE ACTIVIDADES HECHAS POR CADA AGENTE
- MEJORAR ACCION DEL GRUPO:
  - **AUMENTO DE RENDIMIENTOS**
  - **DISMINUCION DE CONFLICTOS**

# COORDINACION DE ACCIONES

- POR QUÉ COORDINAR:
  - AGENTES NECESITAN RESULTADOS E INFORMACION PRODUCIDAS POR OTROS AGENTES
  - RECURSOS SON LIMITADOS
  - OPTIMIZAR COSTOS
  - PONER A TRABAJAR JUNTOS A AGENTES CON OBJETIVOS DIFERENTES PERO QUE DEPENDEN ENTRE SI

# COORDINACION DE ACCIONES

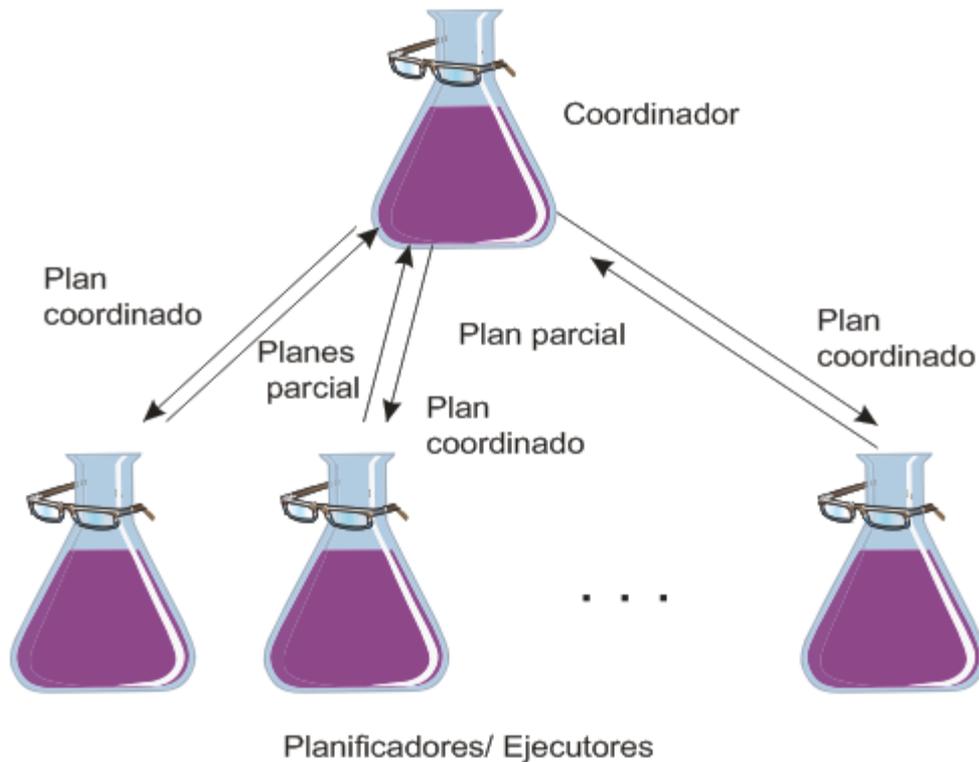
Dos enfoques para su solución:

- **Centralización**
  - Un agente coordinador especial es el responsable de detectar las interdependencias entre las actividades de los agentes locales.
- **Descentralización**
  - Los agentes interactúan entre sí. Poseen el conocimiento para descubrir inconsistencias entre sus acciones previstas y adaptar mutuamente sus decisiones locales.

# FORMAS DE COORDINACION DE ACCIONES

- **SINCRONIZACION:**
  - DE MOVIMIENTOS
  - DE ACCESO A RECURSOS => SECCIONES CRITICAS
- **POR PLANIFICACION**
  - PLANIFICACION CENTRALIZADA VS. DISTRIBUIDA
  - PLAN CENTRAL VS. DISTRIBUIDO
- **REACTIVA**
  - ACCIONES SITUADAS (CAMPOS DE POTENCIAS , RECURSO=>SINCRONIZACION)
  - MARCANDO EL AMBIENTE
  - DE MANADA/O JAURIA (AGREGAR, DISPERSAR, EVITAR, VIGILAR)
- **POR REGLAMENTACION**
  - REGLAS SOCIALES

# Planificación



Planificación centralizado para planes distribuidos

Coordinación centralizada para planes parciales

Planificación distribuida de un plan centralizado

Planificación distribuida para planes distribuidos

Jerarquización de planes.

Planificación reactiva

# Negociación

# Conflictos

- Los conflictos surgen cuando al resolver un problema se da una o varias de las siguientes circunstancias:
  - Coexisten metas diferentes y divergentes en algún momento.
  - Hay diferentes criterios de evaluación de soluciones.
  - Los recursos son limitados.
- Aspecto positivo:
  - se intercambia información
  - mejora la robustez e integración
  - se llega a soluciones globalmente óptimas.
- Los conflictos se resuelven o evitan mediante mecanismos de:
  - prevención y evitación (clásicos en S.O.),
  - sistemas de pizarra (sincronización),
  - negociación.

# Negociación

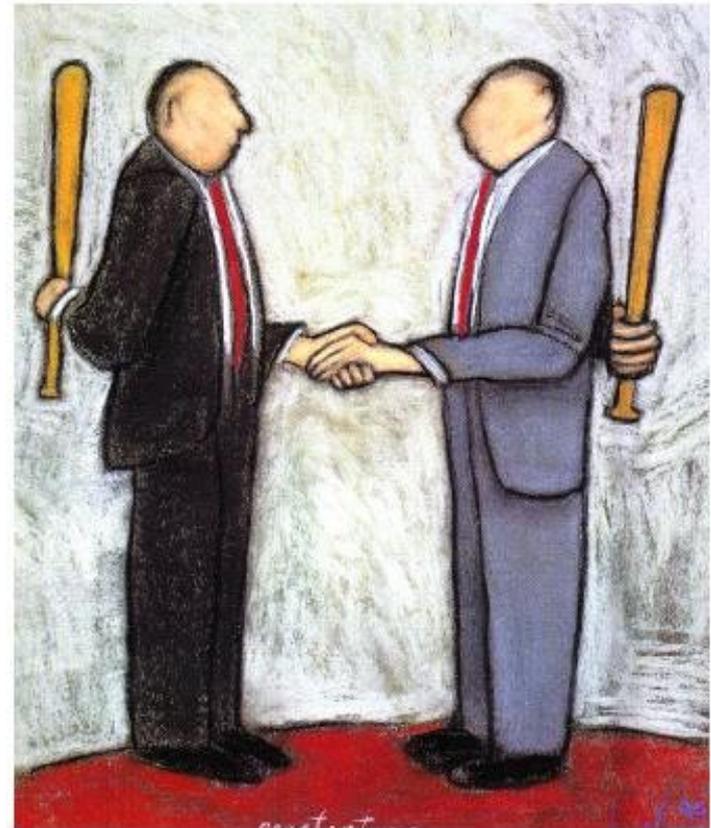
Forma de interacción en la cual un grupo de agentes, con intereses conflictivos y un deseo de cooperar, intentan alcanzar un acuerdo mutuamente aceptable en la división de recursos escasos (Rahwan et al., 2003a)

*La negociación se puede resolver con un plan común*

- **Negociación estricta:**
  - Negociación de un plan común;
  - Ejecución estricta del plan.
- **Otros aspectos de la Negociación:**
  - Existen esquemas conocidos de negociación y relajación de metas
  - Existen mecanismos de arbitraje bien conocidos

# Negociación

- **Objetivo:**  
determinar (las condiciones de) un acuerdo entre, al menos, dos agentes
- **Ejemplos de negociación**
  - **Red de Contratos**
    - Adjudicar productos y tareas a través de un “mercado”
    - $n$  participantes, transacción entre 2
  - **Regateo**
    - Llegar a un acuerdo entre *todos* los participantes
  - **Argumentación**
    - Resolver (supuestos) conflictos a través del debate





# Redes de Contrato

- Basado en mecanismos de mercado usando precios globales y un único mediador centralizado.
- Un único mediador puede convertirse en un cuello de botella (comunicación y computación) o un punto de caída potencial.
- Un contratado se dice individualmente racional (IR) si el agente está mejor con el contrato que sin él.
- Un contratado acepta un contrato si el beneficio es mayor que el costo marginal de gestionar las tareas del contrato.

# Subastas

- Las subastas suelen establecer un contrato entre dos agentes (el subastador y el que gana la subasta).
- Mecanismo **estructurado** para forjar acuerdos
  - Protocolo: semi-distribuido, con diferentes roles
    - 1 subastador
    - $N$  subasteros
  - Estrategias:
    - “pujas” de los subasteros
    - Precio inicial, precio de reserva, etc.

# Elementos de un Protocolo de Subasta

- Tipo de ofertas:
  - abierto (*open-cry*): los subasteros conocen mutuamente sus ofertas
  - privado/cerrado (*sealed-bid*): los subasteros sólo conocen sus propias ofertas
- Proceso de ofertas:
  - una vuelta (*one-shot*): los subasteros sólo dan una oferta
  - directa (*forward*): el precio de las ofertas va ascendiendo
  - inversa (*reverse*): el precio de las ofertas va descendiendo
- Proceso de adjudicación:
  - ¿Qué oferta se usa para determinar el precio que ha de pagar el ganador? (*first-price, second-price, ...*)

# Subasta

Aparece en numerosas aplicaciones en informática.

Oferta 1  
Agente 3



Oferta 2  
Agente 1



Oferta 3  
Agente 3



Precio inicial

Precio privado

Oferta ganadora

## Protocolos comunes:

Subasta inglesa

Subasta holandesa:

Subasta sellada:

•  
Subasta de Vickrey

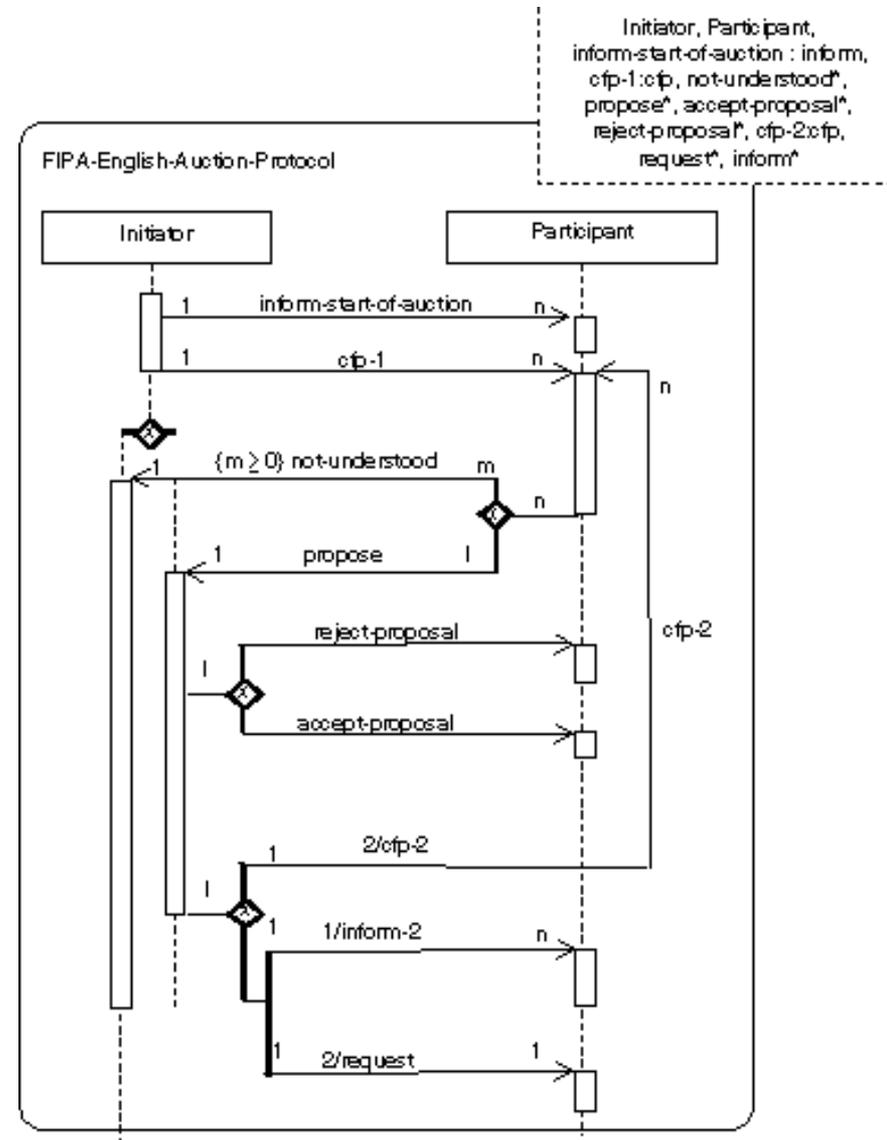
...

# Subasta inglesa

- Inicio:
  - el subastador ofrece un producto a un **precio inicial** (usualmente por debajo de un precio mínimo privado)
- Apuestas:
  - los subasteros van ofertando precios (ninguna, una, o varias veces)
  - cada oferta tiene que **superar** todas las anteriores
  - el ciclo de apuestas termina cuando no hay más ofertas
- Adjudicación:
  - si la última oferta alcanza el precio mínimo (privado) del subastador, el producto es adjudicado al subastero de la oferta *más alta*
  - de lo contrario no se vende el producto (**el subastador tiene la última palabra!!!**)

# Subasta inglesa entre agentes: FIPA

- En la subasta de agentes, los participantes *no* están físicamente presentes en una sala de subasta
- **reject-proposal**: pueden llegar pujas ilegales, p.ej. por retardos en la red
- **cfp**: anunciar cada nueva ronda de pujas con el precio actual
- **inform**: informar a todos los participantes sobre el resultado de la subasta
- **request**: requerir que el ganador realice la transacción





# Comunicación

# Comunicación

Es la intención de intercambiar información a través de la producción y percepción de signos en un sistema compartido de signos convencionales

- Comunicación entre agentes
  - Percepción: Recibir mensajes
  - Acción: Enviar mensajes

# Comunicación de Agentes

- La comunicación entre agentes permite la interacción e interoperación entre agentes, es decir sincronizar acciones, enviar y recibir conocimiento, resolver conflictos en la resolución de una tarea, etc.
- La comunicación permite a los agentes coordinar acciones y comportamientos para dar lugar a sistemas coherentes, capaces de conseguir las metas propias de los agentes o globales del sistema
- Los agentes que cooperan en la realización de tareas tienen un comportamiento social
  - Los agentes intercambian conocimiento: además de datos, un mensaje tiene asociado un contenido semántico
  - Los agentes realizan “actos de comunicación”: Un agente influye en el conocimiento y acciones de otros agentes a través de un acto de comunicación



# COMPONENTES COMUNICACION

## EMISOR

- INTENCION
- GENERACION
- SINTESIS

## RECEPTOR

- PERCEPCION
- ANALISIS
- MANEJO DE AMBIGUEDAD
- INCORPORACION

# COMUNICACION

## Directa:

- Percepción
  - Recibir mensajes
- Acción
  - Enviar mensajes

## Indirecta:

- Percepción
  - Leer del ambiente
- Acción
  - Dejar señal en el ambiente

# PARA QUÉ SIRVE LA COMUNICACION

---

- PARA EXPRESAR
- PARA ORDENAR O SOLICITAR
- PARA REFERENCIAR EN UN CONTEXTO
- PARA VERIFICAR O CONSTATAR
- PARA DARLE VALOR A UN MENSAJE
- META-LINGÜÍSTICA: HABLAR DE OTROS MENSAJES, DE LA SITUACION DE COMUNICACIÓN, ETC.

# ACTOS DE HABLA

*DEFINEN EL CONJUNTO DE ACCIONES  
INTENCIONALES EFECTUADAS EN EL  
TRANSCURSO DE UNA COMUNICACIÓN*

*PRAGMATISMO DEL DISCURSO*

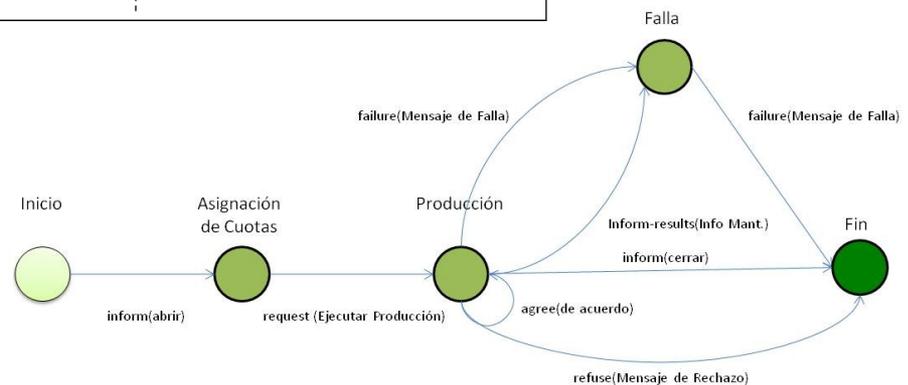
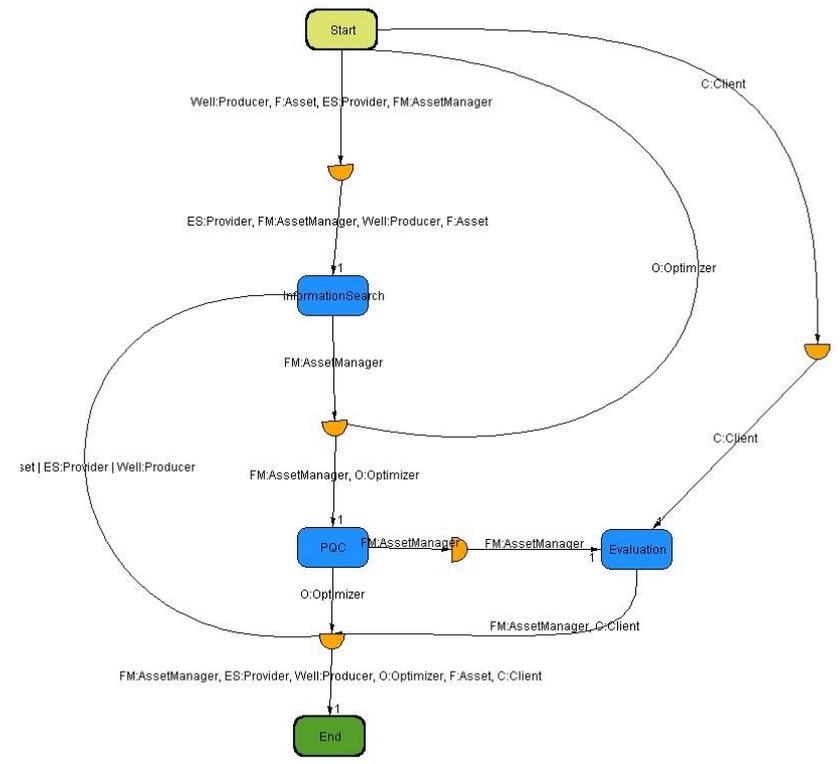
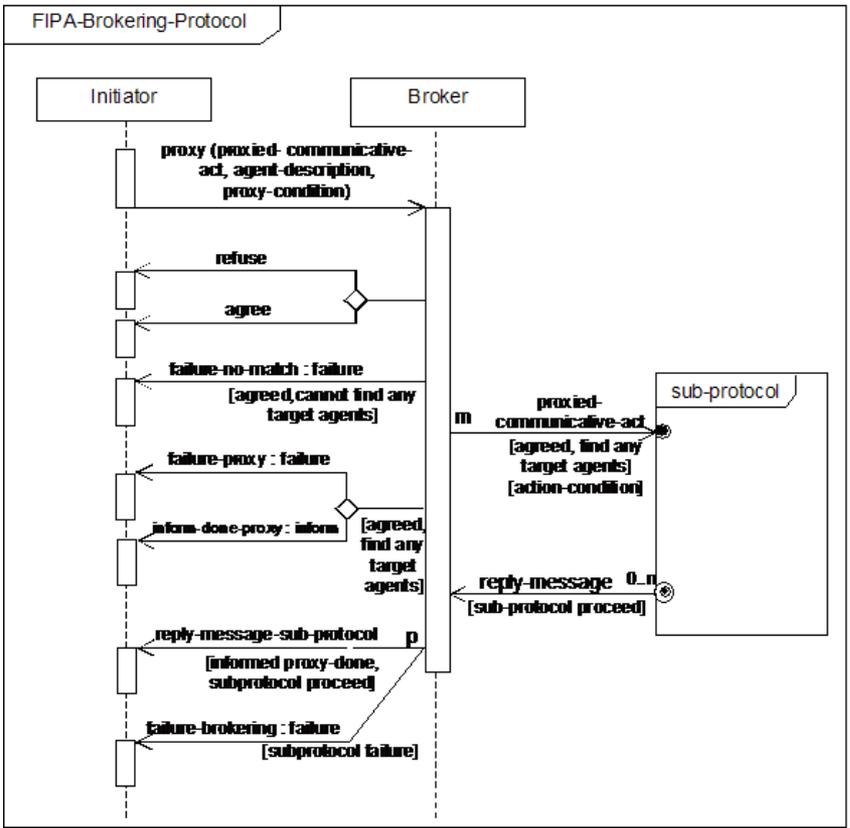
# TIPOS DE ACTOS DE HABLA

<u>TIPO</u>	<u>EJEMPLO</u>	<u>CATEGORIA</u>
DEMANDA	DEMANDA-HACER	ORDENAR
INTERROGACION	PREGUNTAR	INTERROGACION
AFIRMACION	AFIRMAR	ASEGURAR
OFERTA DE SERVICIO	OFRECER-SERVICIO	PROMETER
INDICAR HABILIDADES	SABER-HACER	EXPRESIVO
PROPOSICION HIPOTESIS	PROPONER-IDEA	EXPRESIVO

# CONVERSACIONES

- Patrón para el intercambio de mensajes entre dos o más agentes que acuerdan comunicarse entre si
- Protocolos de conversación: los agentes no realizan un simple intercambio de mensajes aislados, sino que mantienen conversaciones (Se espera cierto tipo de respuesta del interlocutor)
- Ventajas asociadas a los protocolos de conversación
  - Proporcionan un marco para el intercambio de mensajes (interpretación).
  - Posibilitan mejor ajuste a modelos intuitivos de interacción entre agentes.
  - La estructura conversacional es separable del resto del comportamiento del agente (favorece la reutilización)

# CONVERSACIONES



# Lenguajes de Comunicación de Agentes

- **Sintaxis:** Un conjunto predefinido de símbolos.
- **Semántica:** que describen los símbolos
- **Pragmática:** como se interpretan los símbolos

Ejemplos de lenguajes: **KQML, ACL**

# ACL: Agent Communication Language

- Lenguaje para intercambio de mensajes entre agentes.
- Componentes de base
  - Vocabulario
  - Lenguaje “interno”: KIF
  - Lenguaje “externo”: KQML
- Un mensaje en ACL es una expresión **KQML**, en la que los argumentos son términos o sentencias **KIF**, formadas por palabras del **vocabulario** del ACL

# ACL: Agent Communication Language

- El mensaje ACL posee los campos:
  - sender: agente que envía el mensaje.
  - receivers: lista de receptores.
  - performative: tipo de mensaje. Indica la intención que el emisor intenta lograr enviando el mensaje.
  - content: contenido principal del mensaje.
  - language: lenguaje usado en el contenido. P.ej. la sintaxis utilizada para expresar el contenido.
  - ontology: ontología usada en el contenido. P.ej. el vocabulario de símbolos usados en el contenido y su significado.
  - conversation-id, reply-with, in-replay-to, reply-by: para el control de conversaciones concurrentes..

# KQML

- KQML es un lenguaje y protocolo basado en mensajes para comunicación entre agentes.
- Los mensajes KQML no sólo comunican frases en algún lenguaje sino que mas bien comunican una *actitud* sobre el contenido
- Las primitivas del lenguaje reciben el nombre de “*performatives*”
- Así, cada mensaje posee
  - Performativa
  - Parámetros

# KQML

- Tres capas
  - Contenido (KIF)
  - Mensaje (Performativas)
  - Comunicación
- Performativas
  - De discurso: para intercambio de información y conocimiento (ask, tell)
  - De intervención y conversación: para intervenir en una conversación (sorry, next, ready)
  - De red: no son estrictamente actos de habla, pero permiten a los agentes encontrar otros agentes (recruit, register)

# KQML

- Mensajes en KQML

```
( Performativa
```

```
    : content (<Contenido a transm.>)  
    : receiver <destino>  
    : language <lenguaje del contenido>  
    : ontology <ontologia usada>
```

```
)
```

- Ejemplo

```
( ask-one
```

```
    :content (PRICE IBM ?price)  
    :receiver -stock-server  
    :language LPROLOG  
    :ontology NYSE-TICKS
```

```
)
```

# KQML (Realizativas o performativas)

- Se usan para los actos comunicativos (Actos de Habla)

## Actos Comunicativos

## Operaciones

para preguntas básicas

evaluate, ask-if, ask-in, ask-one, ask-all

de preguntas con varias resp.

stream-in, stream-all

de respuesta

reply, sorry

de información genérica tell, achieve, cancel, untell ...

para control de flujo

standby, ready, next, rest, discard ...

que define capacidades advertise, suscriber, monitor, export ...

de gestión

register, forward, broadcast, router ...

# KIF

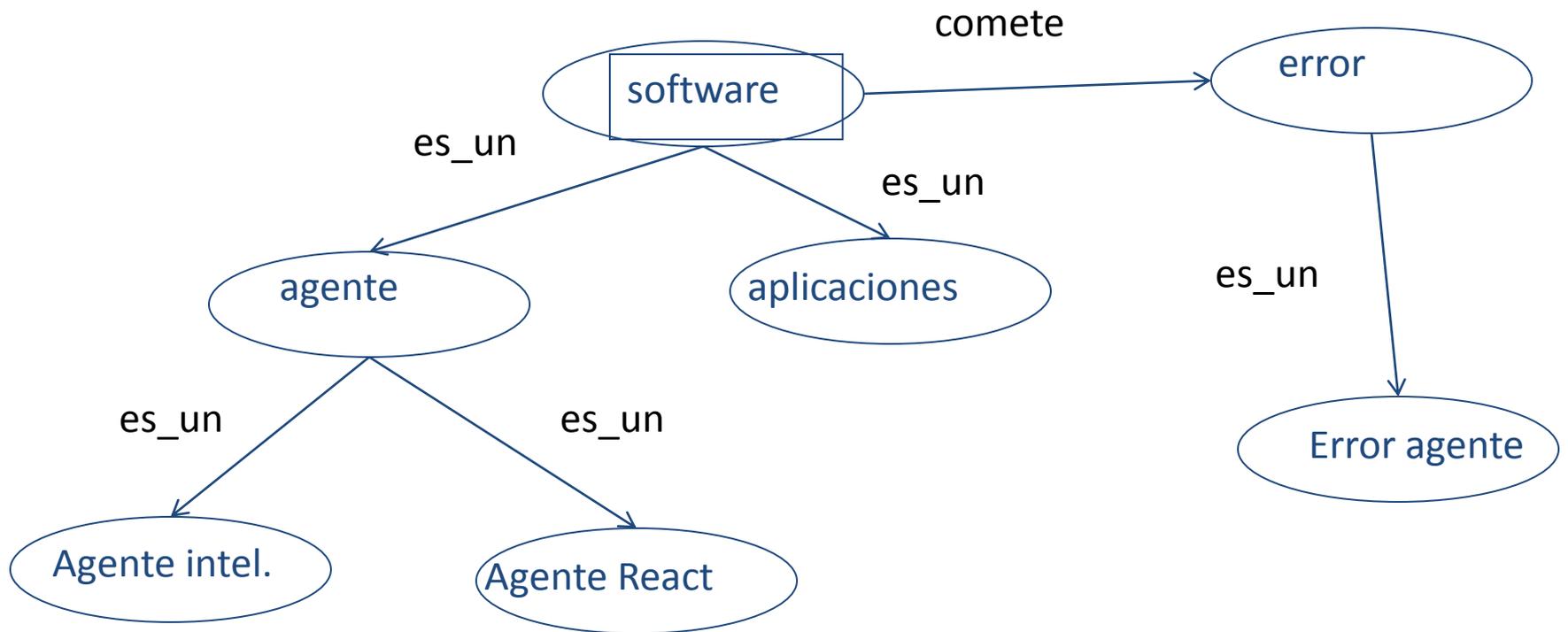
- Originalmente desarrollado con la intención de ser un lenguaje común para expresar propiedades de un dominio en particular.
- Lenguaje de predicados de primer orden
- Notación prefija (LISP)
- Ejemplo
  - (= (temperature ml) (scalar 83 Celsius))
  - (defrelation bachelor (?x) :=  
    (and (man ?x)  
      (not (married ?x))))

# Ontologías para la comunicación entre agentes

- Uso habitual de las ontologías  
para lograr la compartición y la comunicación del  
conocimiento:
- Necesidad de acordar terminología.
  - Lenguaje de representación de conocimiento común
  - Lenguaje de comunicación común (lenguaje de  
comunicación entre agentes)
- La semántica es muy importante:
  - Los agentes se comunican para entenderse
  - necesitan ontología común o reglas de traducción

# Ontologías para la comunicación entre agentes

“El agente cometió un error.”



# FIPA (Foundation for Intelligent Physical Abstracts)

# FIPA (Foundation for Intelligent Physical Abstracts)

- FIPA es una organización de la IEEE Computer Society que promueve la tecnología basada en agentes y la interoperabilidad de sus estándares con otras tecnologías.
- FIPA es la organización estándar para agentes y sistemas multi-agentes, fue oficialmente aceptada por la IEEE en Junio del 2005.
- FIPA fue originalmente creada en Suiza en 1996 para producir especificaciones de estándares de software para agentes y sistemas basados en agente heterogéneos e interactivos.

# Agentes: FIPA

- La FIPA ha desarrollado una serie de especificaciones, las cuales son una colección de estándares :
  - Marcos arquitectónicos para SMAs,
  - Protocolos de comunicación,
  - Lenguajes de contenido,
  - Mecanismos de gestión e implantación de SMAs,
  - Estándares de calidad de servicio, entre otras.

# Agentes: FIPA

- Las especificaciones están divididas en 5 grupos:
  - Aplicaciones:
    - FIPA Nomadic Application Support Specification
    - FIPA Quality of Service Specification
    - FIPA Personal Travel Assistance Specification
    - FIPA Audio-Visual Entertainment and Broadcasting Specification
    - FIPA Network Management and provisioning Specification
    - FIPA Personal Assistant Specification
    - FIPA Message Buffering Service Specification
  - Arquitectura Abstracta
    - FIPA Abstract Architecture Specification
    - FIPA Domains and Policies Specification

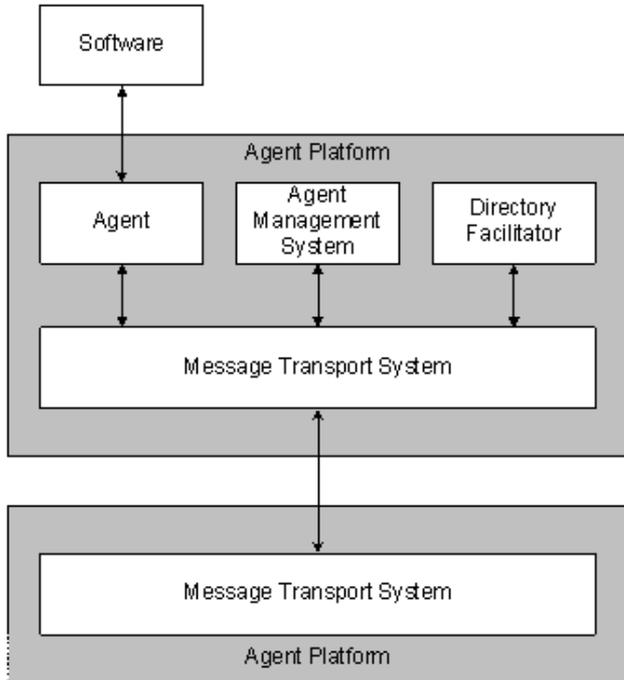
# Agentes: FIPA

- Lenguajes de comunicación
  - Protocolos de interacción
    - FIPA Request Interaction Protocol Specification
    - FIPA Query Interaction Protocol Specification
    - FIPA Request When Interaction Protocol Specification
    - FIPA Contract Net Interaction Protocol Specification
    - ...
    - FIPA Brokering Interaction Protocol Specification
    - FIPA Recruiting Interaction Protocol Specification
    - FIPA Subscribe Interaction Protocol Specification
    - FIPA Propose Interaction Protocol Specification
    - FIPA English Auction Interaction Protocol Specification
    - FIPA Dutch Auction Interaction Protocol Specification

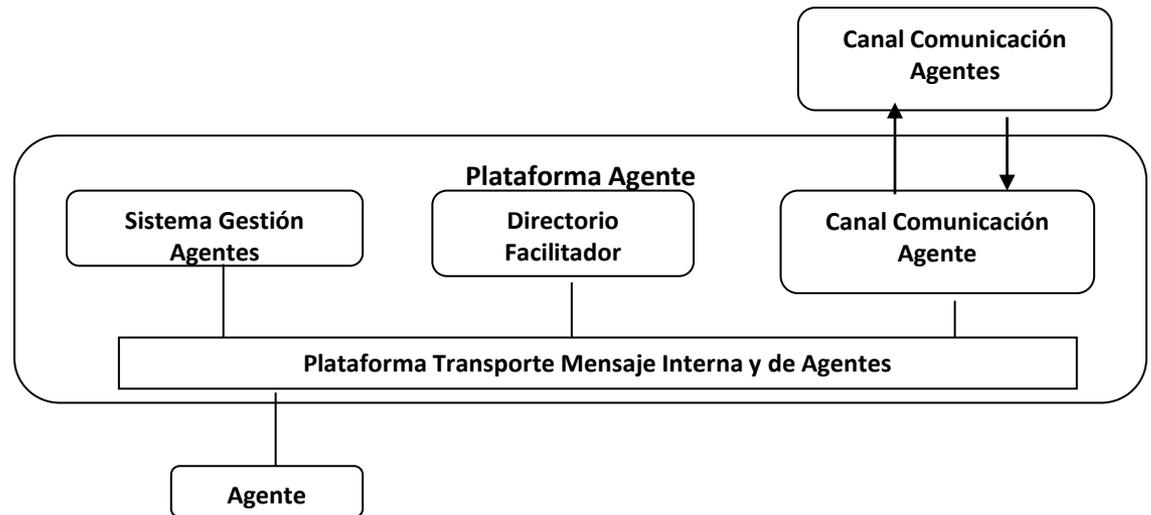
# Agentes: FIPA

- Actos comunicativos
  - FIPA Communicative Act Library Specification
- Lenguajes de contenido
  - FIPA SL Content Language Specification
  - FIPA CCL Content Language Specification
  - FIPA KIF Content Language Specification
  - FIPA RDF Content Language Specification
- Gestión de agentes
  - FIPA Agent Management Specification
  - FIPA Agent Discovery Service Specification
  - FIPA JXTA Discovery Middleware Specification

# Agentes: FIPA



Estándar de plataforma  
de servicios





UNIVERSIDAD  
DE LOS ANDES  
MÉRIDA VENEZUELA

# Retos en los SMA

# Retos

- Problema de la confianza
- Aprendizaje en SMA
- Auto-organización

# La Confianza en los SMA

- El agente Ag1 **cre**e que el agente Ag2 puede y desea hacer una acción dada;
- El agente Ag1 **tiene la meta** de que el agente Ag2 haga esa acción; y
- El agente Ag1 **confía** en el agente Ag2, absteniéndose de ejecutar la acción delegada y **coordinando su comportamiento con el comportamiento esperado** del agente Ag2.

# La Confianza en los SMA

- Dificultad de modelización
  - Medida de la **estima** o del **prestigio**
  - **Subjetividad**
    - ¿Qué es algo bueno? ¿Para quién?
- Dificultad para el diseño
  - Medidas **cuantitativas** o **cualitativas**...
    - $t_{A \rightarrow B} = 0.7$  |  $t_{B \rightarrow A} = 0.3$  |  $t_{B \rightarrow C} = \text{“bueno”}$  |  $t_{B \rightarrow A} = k$
- Distintos campos confluyen: **informática** (modelos computacionales), **sociología** (redes sociales), **psicología** (estados mentales), **economía** (funciones de utilidad, toma de decisiones, etc.), ...

# Modelos de confianza

- Clasificación de los modelos de confianza
  - Modelos basados en Aprendizaje y Evolución
  - Modelos basados en Reputación
  - Modelos Socio-Cognitivos
  - Modelos para Organizaciones
  - Alineación semántica

# Modelos de confianza basados en Reputación



- Agregación
  - Combinación de valores de reputación que llegan a un agente
  - Ejemplo: eBay (+1, -1). Comentarios?
    - Valoraciones detalladas del vendedor con cuatro categorías diferentes.
    - Los compradores se les pidió que calificaran el vendedor en cada una de estas categorías con una puntuación de una a cinco estrellas, siendo cinco la calificación más alta y uno más bajo.
    - Estas calificaciones son anónimas.
    - Las listas de los vendedores con una calificación de 4.3 o más en cualquiera de las cuatro categorías de calificación aparecen antes en los resultados de búsqueda.
    - Tiendas de alimentación están obligados a tener las puntuaciones de cada categoría por encima de 4,5.

# Aprendizaje de un Agente vs Aprendizaje SMA

---

- Casi todos los algoritmos de *aprendizajes han sido hechos para un agente*

Como usarlos en SMA?

- Algoritmos de aprendizajes para un agente se focaliza en como un agente mejora sus habilidades individuales.
  - No podemos hablar de aprendizaje SMA, si un agente no afecta ni es afectado por otros agentes
- si un agente no es explícitamente consciente de otros agentes, lo percibe como parte del medio ambiente y su comportamiento será parte de la hipótesis a aprender.
- Es posible lograr un comportamiento coordinado del grupo usando aprendizaje para un solo agente

# Aprendizaje de un Agente vs Aprendizaje SMA

---

- Investigaciones anteriores han demostrado que ciertos niveles de conocimiento de los agentes pueden perjudicar el rendimiento.
- El aprendizaje para un agente no siempre produce un rendimiento óptimo en SMA y pueden existir dominios donde *un aprendizaje coordinada multi-agente* es una metáfora más natural y mejora la eficacia.

es una pregunta abierta si niveles altos de conocimiento en un agente producen un mejor desempeño.

# ¿Cuál es el objetivo en el aprendizaje de SMA?

- Debido a que no conoce su entorno
  - ¿Cómo se comportan los otros agentes?
  - ¿Cuál es la función de recompensa?
- Aprender una mejor respuesta
  - Convergencia de políticas.
  - El minimax óptimo.
- Información completa: resuelto
  - Soluciones exactas o aproximadas
- Información Incompleta: **se requiere aprender**
  - El entorno no es markoviano
  - La convergencia no está garantizada
  - Comportamientos impredecibles

# Interés del Aprendizaje en SMA

## Relacionada con la Organización

Aprendizaje de roles

Aprendizaje sobre otros agentes

Aprender a jugar mejor contra un  
oponente

Aprender a beneficiarse de las  
condiciones del entorno

## Relacionada con Coordinación

Aprender a coordinarse evitando fallos

Adaptación a distintas situaciones



# Aprendizaje y Comunicaciones

- Aprender a Comunicarse
- Comunicación como aprendizaje
- Temas:
  - ¿ Qué comunicar?
  - ¿ Cuando comunicar?
  - ¿ Con quien comunicarse?
  - ¿ Cómo comunicarse?

# SMA: auto-organización y emergencia

Auto-organización, auto-adaptación y auto-equilibrio son propiedades fundamentales de los **SMA con propiedades emergentes**, que ninguno de los agentes individuales tiene

- La sinergia tiene que ser considerada una propiedad emergente: una mejor sinérgica hace la diferencia entre la resolución de una tarea o no.
- Otras propiedades específicas a los sistemas auto-organizados
  - toma de decisiones a nivel local,
  - Escalables, flexibles,
  - de baja observabilidad y
  - pobre controlabilidad.

# Ejemplos de auto-organización en los SMA

- Cooperación entre los agentes
- Distribución de tareas y recursos,
- Coordinación de acciones,
- Resolución de conflictos, entre otros.

**Agrupamiento  
de especialistas,**

**Formación de  
Formas  
geométricas**

# Plataformas existentes para el desarrollo y/o el despliegue de agentes

# Plataformas para el despliegue de SMA

- Define modelos, lenguajes y herramientas de implementación, entre otros aspectos, que faciliten el desarrollo e interoperabilidad de los SMA
- *Middleware (medio de gestión de servicios)*
- Ellos deben permitir a los agentes:
  - Sensar eventos, actuar o manipular el ambiente;
  - Comunicarse;
  - Tener nombres que sean únicos para poder ser identificados;
  - Tener mecanismos de búsquedas;
  - Poder cooperar, negociar y coordinarse.

# Herramientas IDE (Integrated Development Environment) para el desarrollo de SMA

- Desde el punto de vista de los diseñadores, la importancia radica en que ellos necesitan apoyo para desarrollar aplicaciones basadas en agentes.
- En particular, en cuatro etapas se requiere de apoyo:
  - En el análisis del dominio del problema.
  - En el diseño de la arquitectura para la solución del problema.
  - En el desarrollo o construcción de una solución funcional para el problema.
  - En la implementación de la solución para el problema (lanzamiento, operación y mantenimiento).

# Ambientes

- Ambientes híbridos
  - *JADE*
  - *FIPA-OS*
  - *JACK*
  - *Multi-AGent Environment – MAGE*
- Plataformas para el despliegue de SMA
  - *MaDKit*
  - *Magentix*
- IDE para agentes
  - *Amine*
  - *AgentTool*
  - *AgentBuilder*

# JADE

- JADE (*Java Agent DEvelopment Framework*) es un software para desarrollar aplicaciones basadas en agentes que sigue las especificaciones de FIPA,
- Incluye:
  - Una plataforma de agentes compatible con FIPA.
  - Un paquete basado en Java para desarrollar agentes.
  - Varias extensiones para ejecutarse en dispositivos con aplicaciones específicas (PDA, smartphone, etc.).
- JADE ha sido totalmente implementado en el lenguaje Java

# JADE

- Para la implantación de SMA, JADE proporciona:
  - Un entorno de ejecución en el que los agentes se ejecutan.
  - Bibliotecas de clases para la creación de agentes mediante la herencia y la redefinición de comportamientos.
  - Un conjunto de herramientas gráficas para el monitoreo y la administración de la plataforma.
- JADE define la Clase *Agent*, la cual es una superclase que permite a los usuarios crear agentes JADE.
- Básicamente, los agentes se implementan como un hilo por agente

# JADE

- Para la implantación de SMA, JADE proporciona:
  - Un entorno de ejecución en el que los agentes se ejecutan.
  - Bibliotecas de clases para la creación de agentes mediante la herencia y la redefinición de comportamientos.
  - Un conjunto de herramientas gráficas para el monitoreo y la administración de la plataforma.
- JADE define la Clase *Agent*, la cual es una superclase que permite a los usuarios crear agentes JADE.
- Básicamente, los agentes se implementan como un hilo por agente

# Ingeniería del Software Orientada a Agentes

# Ingeniería del Software Orientada a Agentes

- Los agentes representan un nuevo nivel de abstracción que puede ser utilizado por los desarrolladores de software para entender, modelar y desarrollar de un modo más natural una clase importante de sistemas distribuidos.
- Las técnicas de desarrollo software habituales no son adecuadas para esta tarea, ya que no son capaces de capturar los aspectos únicos de los SMA:
  - Comportamiento flexible, autónomo, de resolución de problemas
  - Riqueza en sus interacciones
  - Complejidad de la estructura organizacional

# Ingeniería del Software Orientada a Agentes

- **Agent-Oriented Software Engineering (AOSE)**
  - Gaia
  - MaSE
  - Agent UML
  - Prometheus
  - MASINA
- **Ingeniería del Conocimiento Orientada a Agentes**
  - MASCommonKADS
  - DESIRE
  - Cassiopeia
- **Métodos formales orientados a agentes**
  - Métodos formales en AOSE
  - Especificación en Z

# Metodología MASINA

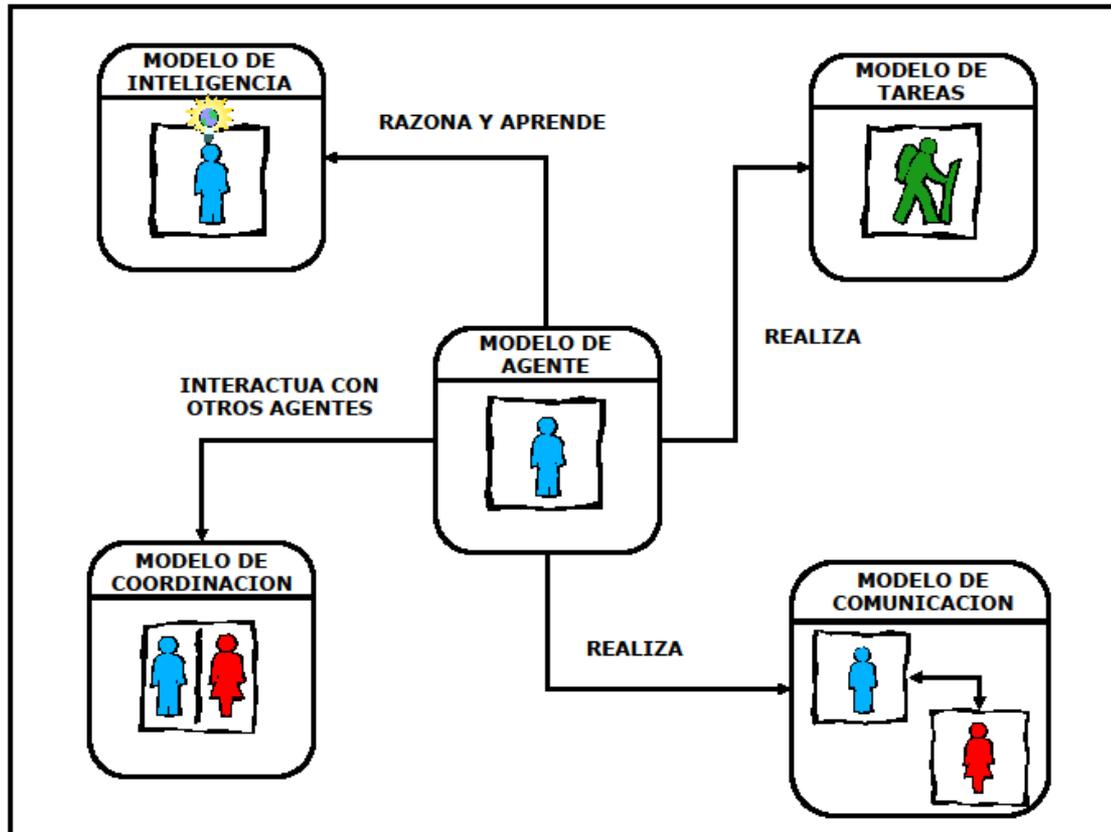
Metodología que permite especificar Sistemas Multi-agentes, la cual es una extensión de MAS-CommonKADS.

## Fases

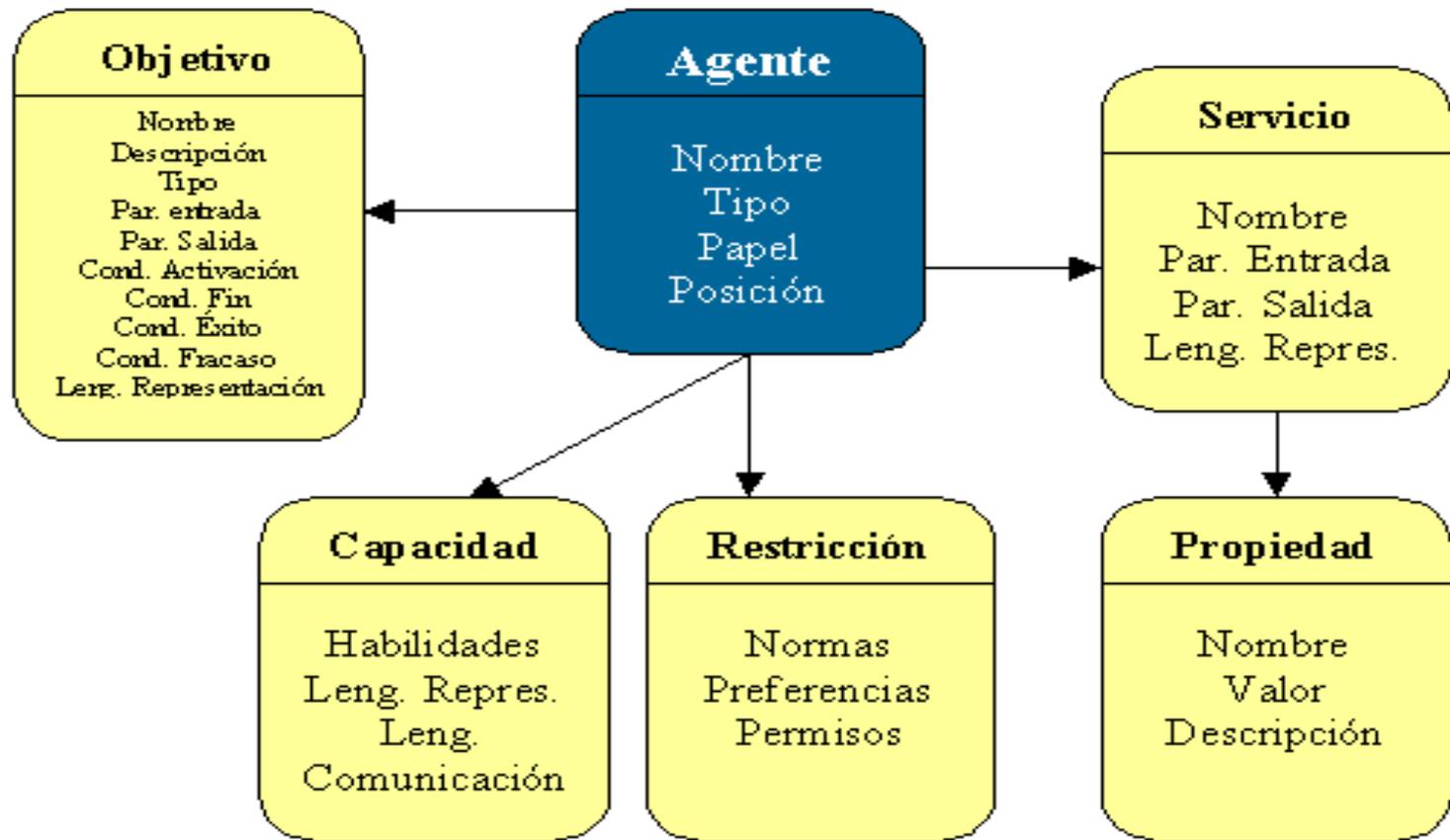
- Conceptualización*
  - Casos de uso (descripción de acciones necesarias para producir un resultado útil)
  - Actores (roles desempeñados por alguna persona, una pieza de software, u otro sistema)
- Análisis y Diseño*
  - Modelos para describir los agentes del sistema, sus tareas, su organización y los medios de comunicación.
  - Diseño técnico del sistema (modelo de implementación).
- Codificación y prueba*
- Integración*
- Operación y mantenimiento*

# MASINA

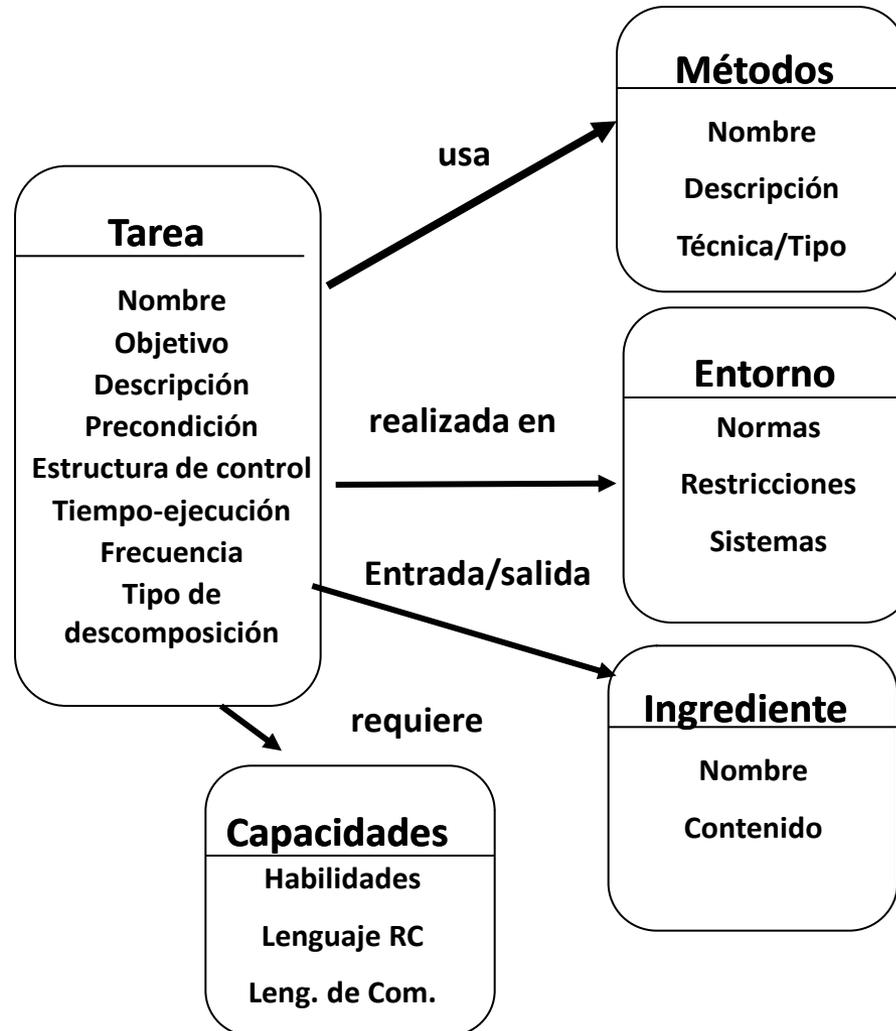
## Fase de Análisis



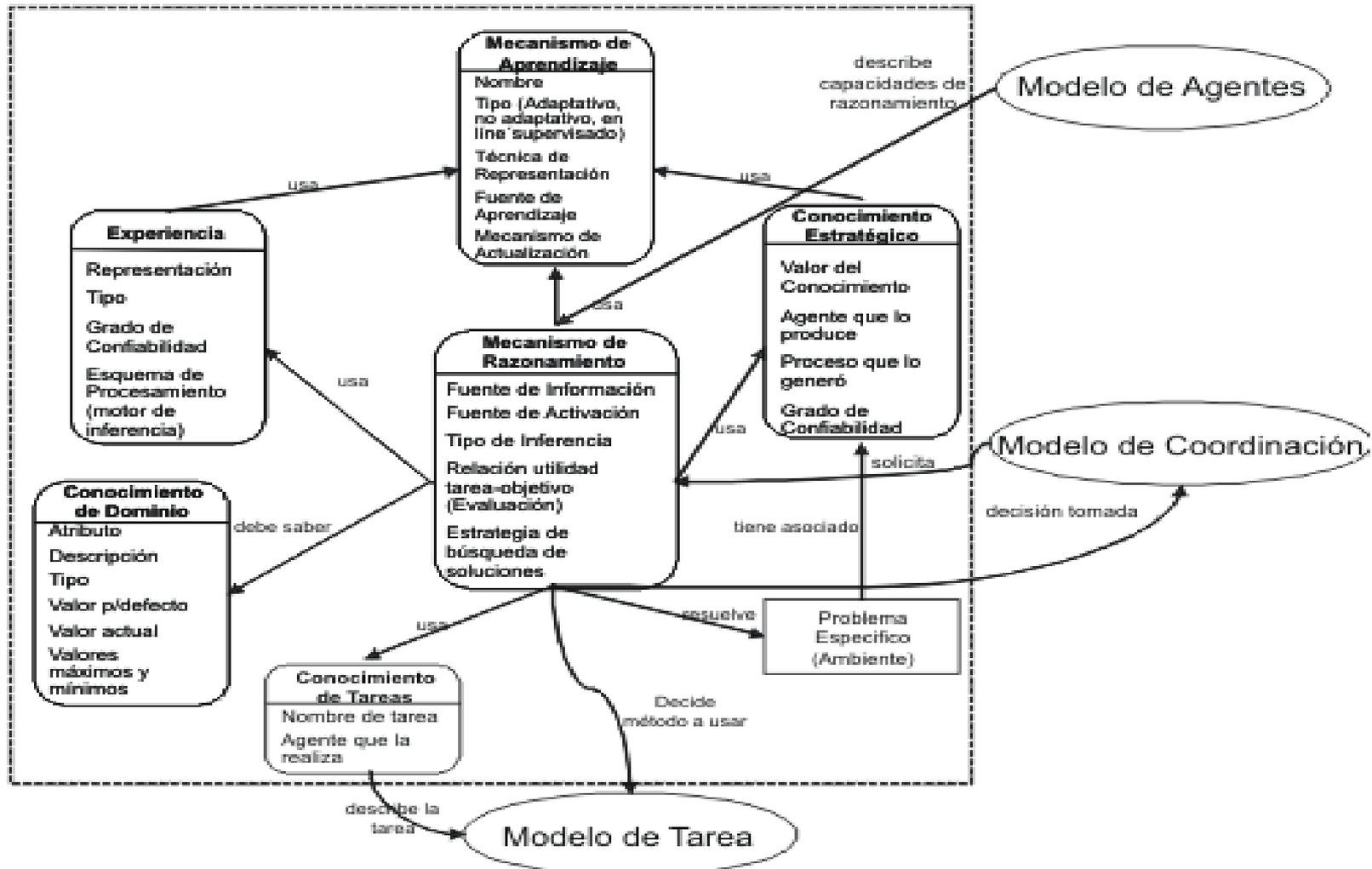
# Modelo de Agente



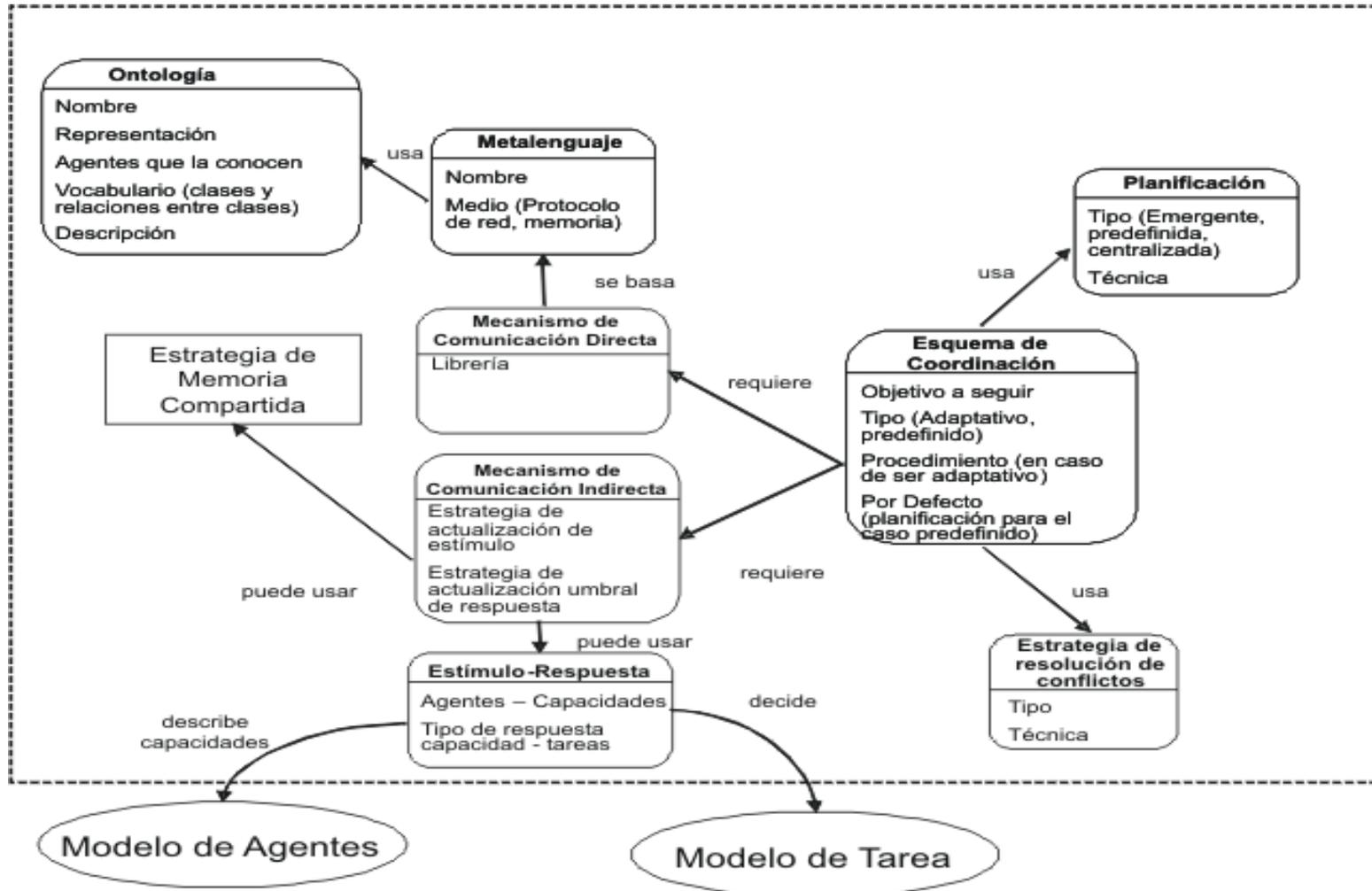
# Modelo de Tareas



# Modelo de Inteligencia



# Modelo de Coordinación



# Modelo de Coordinación y Comunicación

## Conversación

Nombre  
Tipo  
Objetivo  
Agentes  
Iniciador  
Servicio  
Actos de Habla  
Descripción  
Precondición  
Condición de Terminación

## Acto de Habla

Objetivo  
Tipo  
Agentes Participantes  
Comunicación  
Emisor  
Receptor  
Conversación  
Servicio  
Datos Intercambiados  
Descripción  
Precondición  
Condición de Terminación  
Performativa  
Medio de Comunicación

# Plataformas para el desarrollo de Agentes basado en MASINA

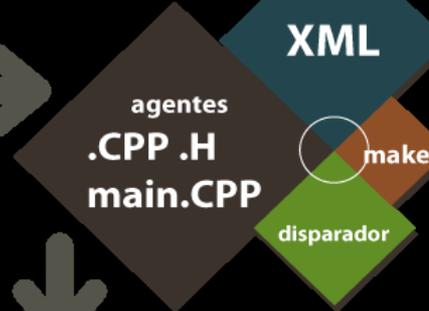
# EDISMA

## AGENTES?

### 1. Diseño



### 2. Digitalizar modelos MASINA



### 3. Completar los Agentes

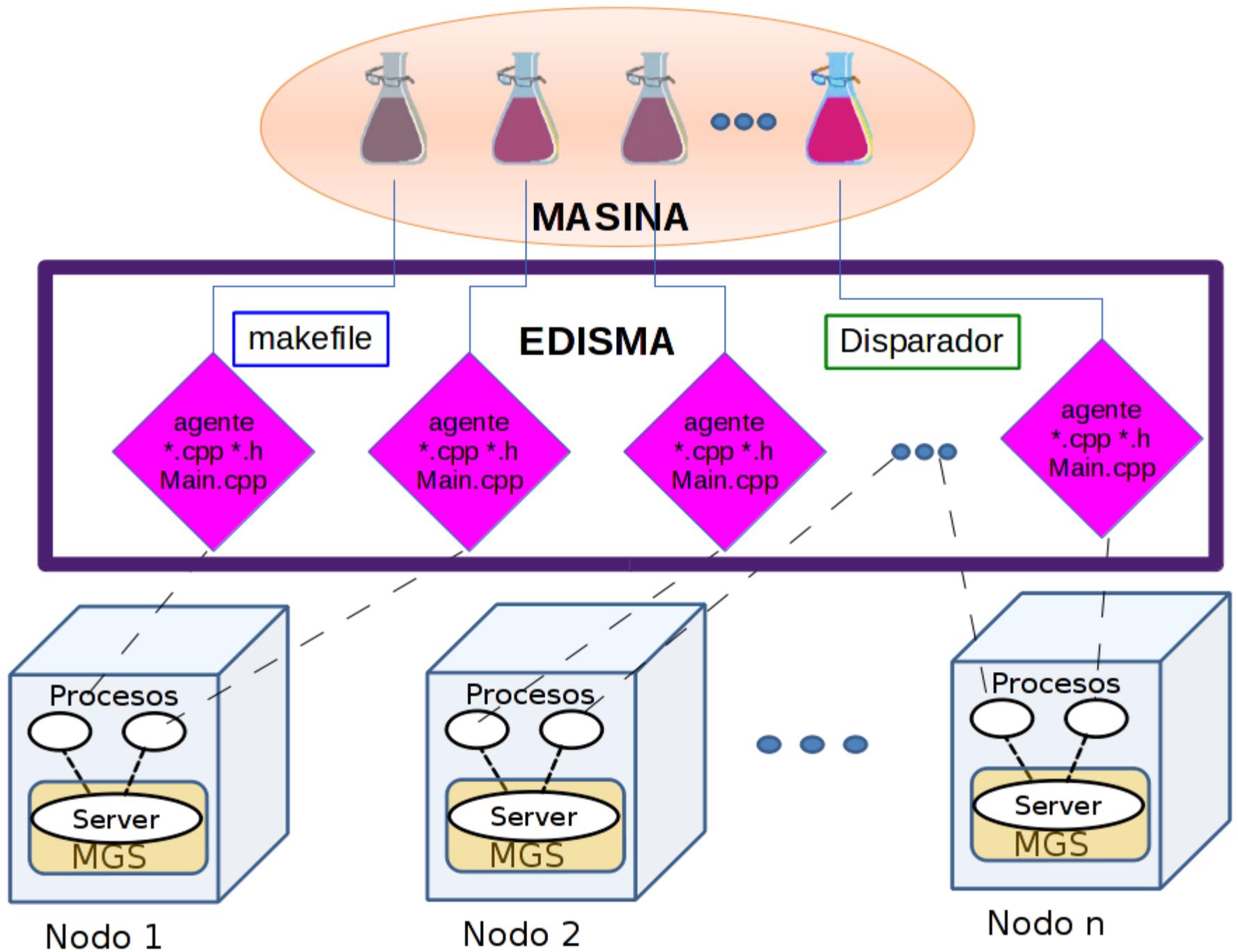


```
#include <agente.h>  
class Agente:  
int atributo1, atributo2;  
int metodo1();
```

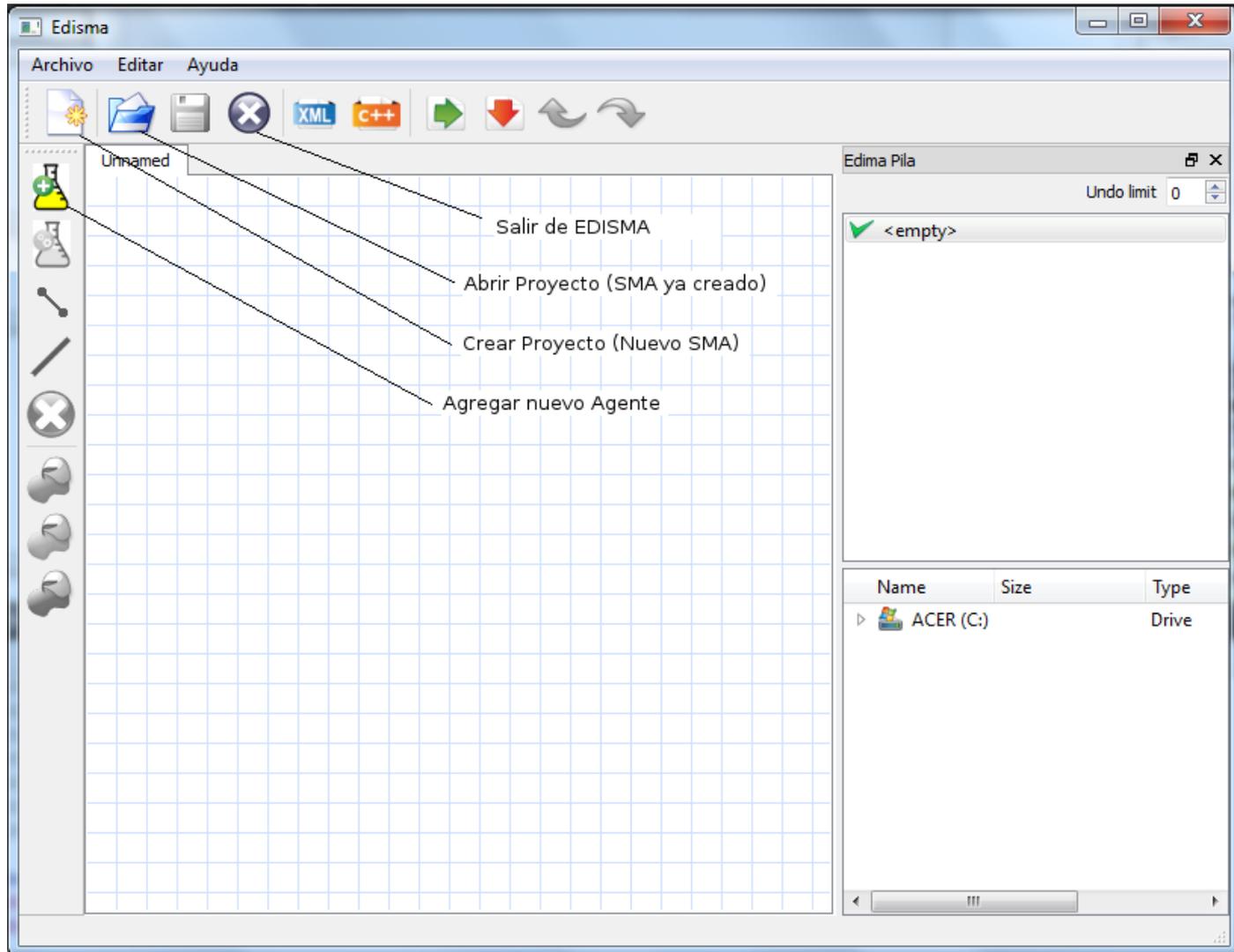


### 4. Compilar y Ejecutar en el MGS



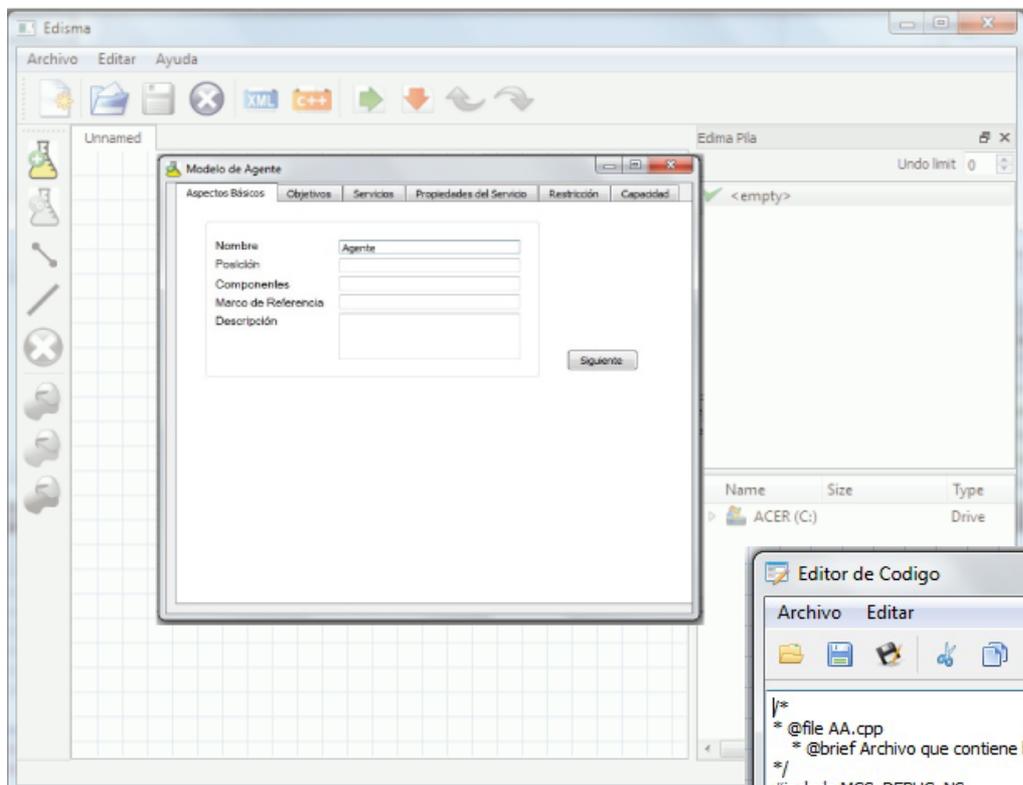


# EDISMA

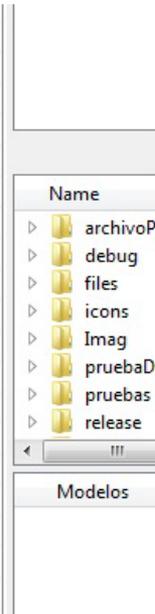
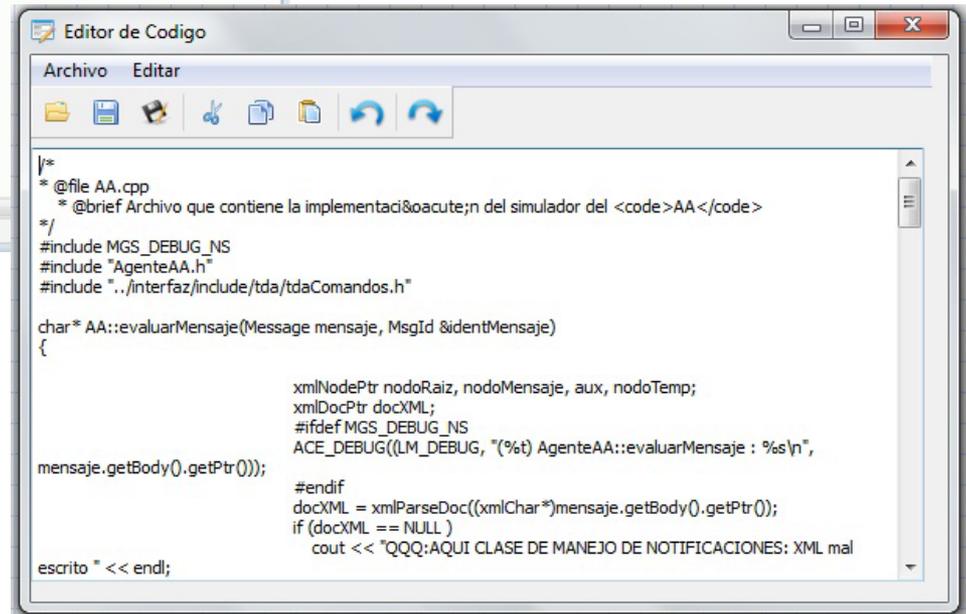


# EDISMA

Modelo de tarea



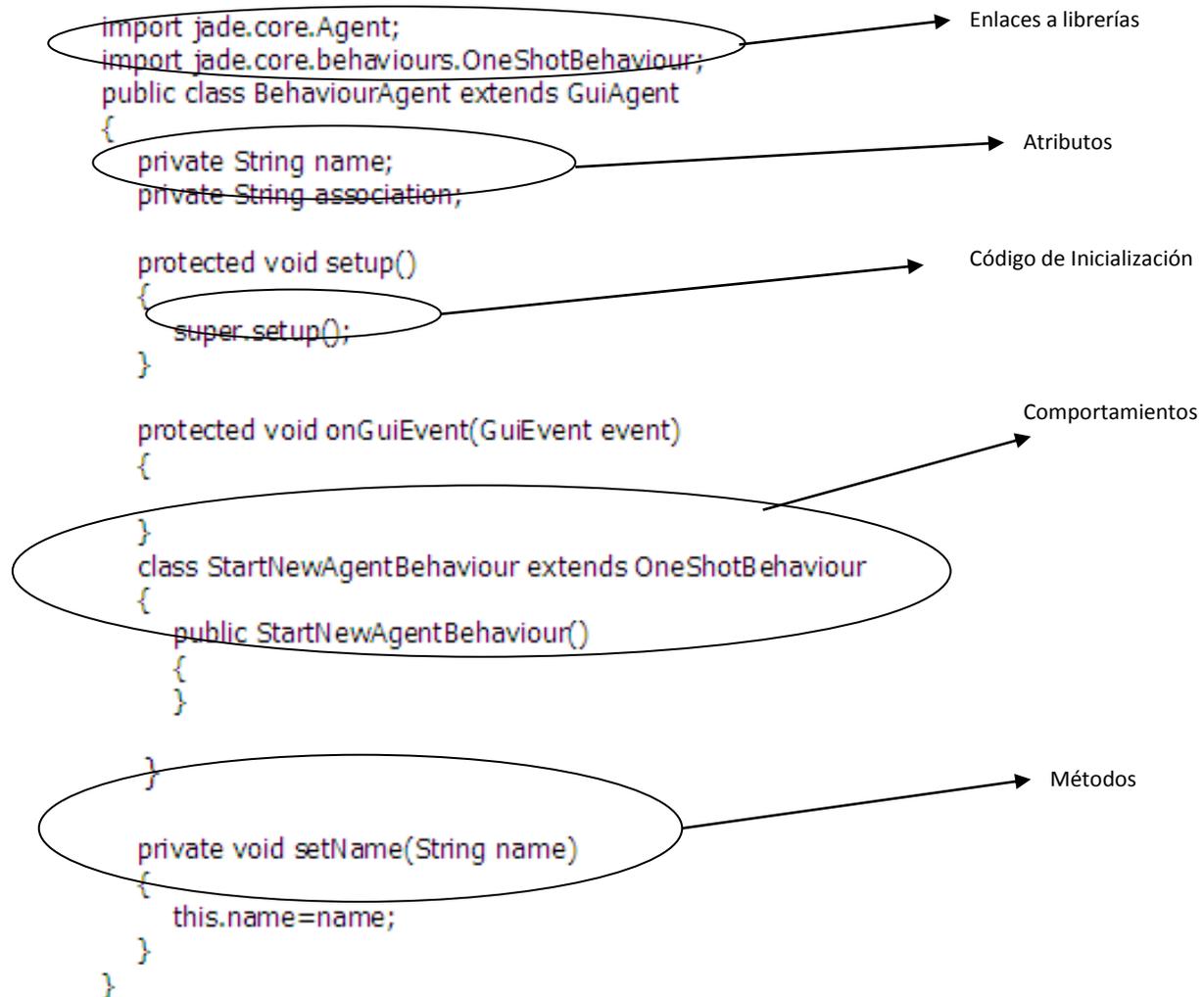
Editor  
código



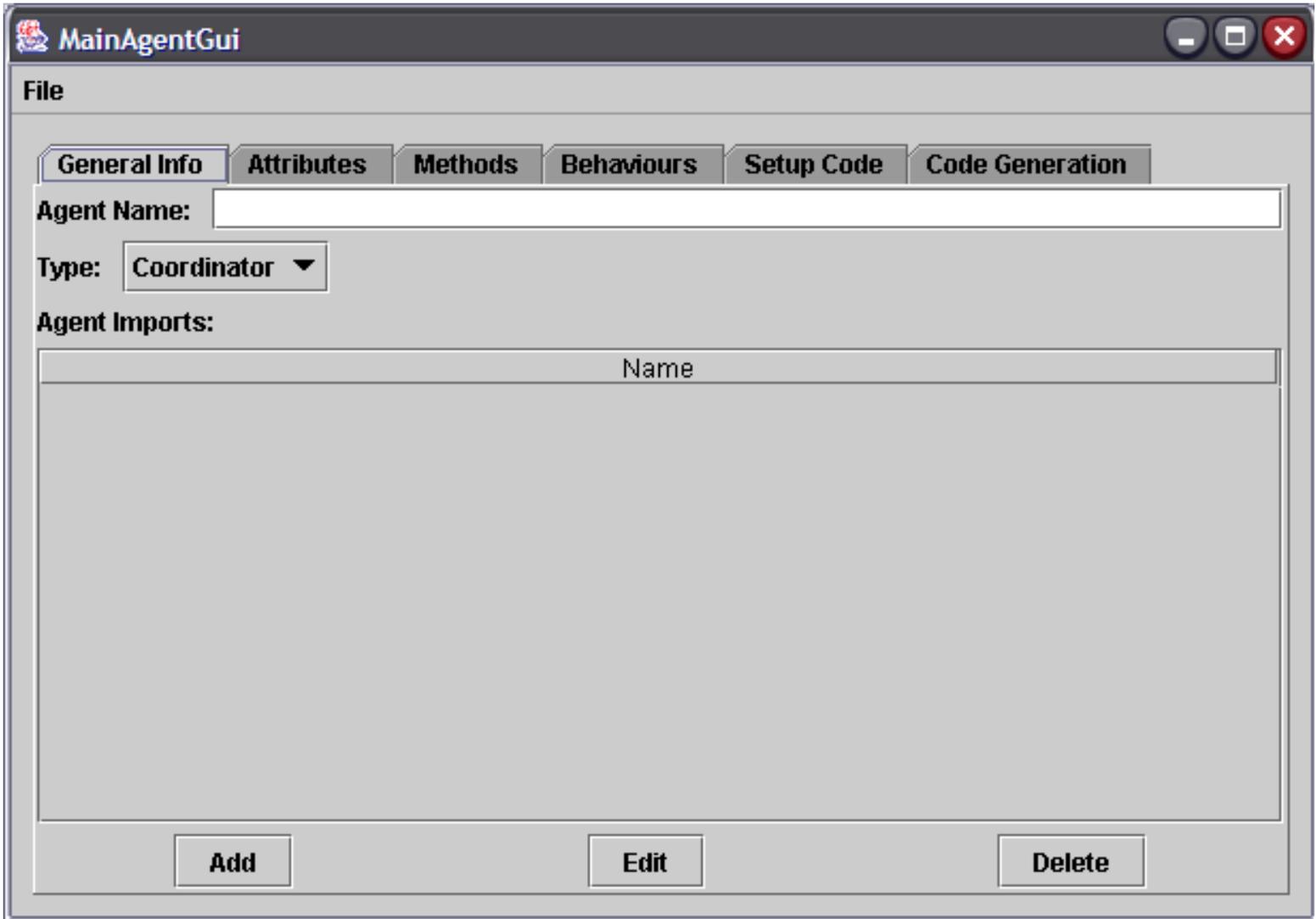
# EDISMA



# Ontología de Generación de código



# DESARROLLO DE SIGECO



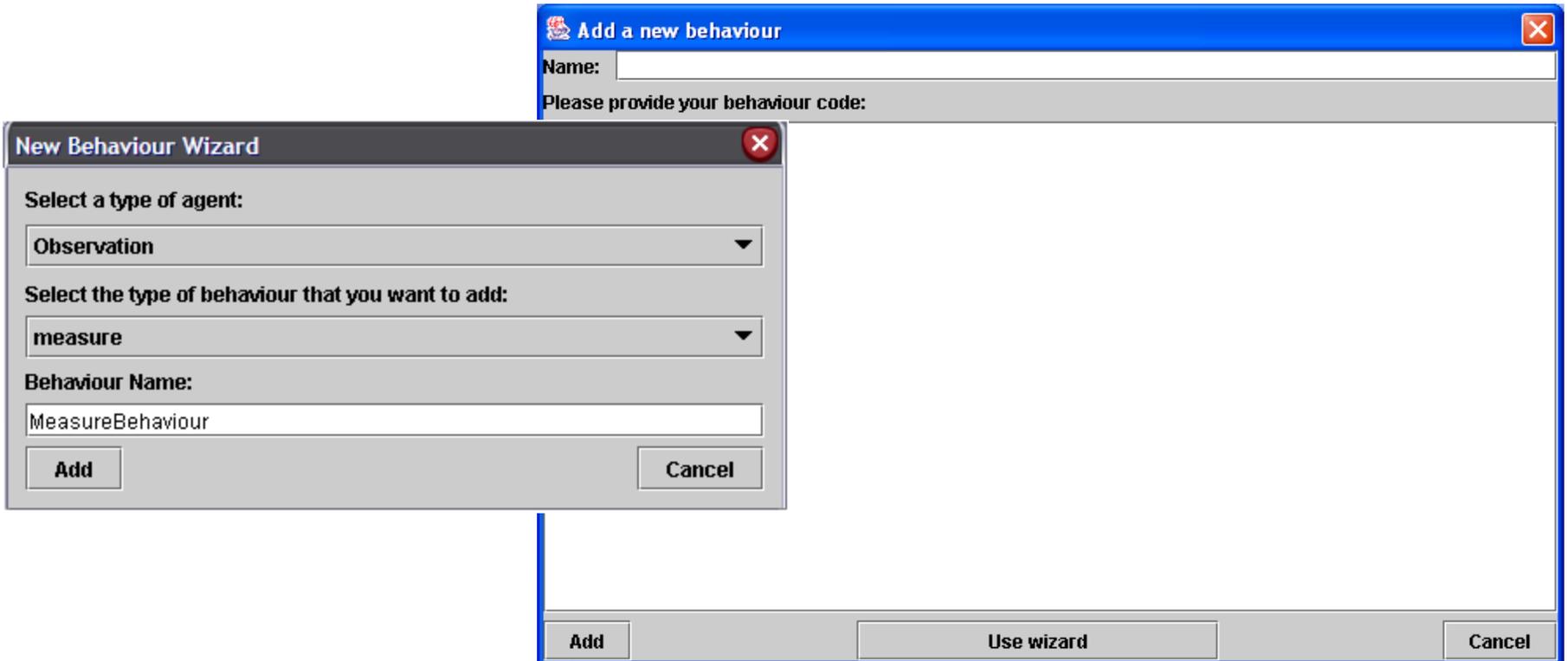
The screenshot shows a software window titled "MainAgentGui" with a standard Windows-style title bar (minimize, maximize, close buttons). The window has a "File" menu and a tabbed interface with the following tabs: "General Info", "Attributes", "Methods", "Behaviours", "Setup Code", and "Code Generation". The "General Info" tab is active and contains the following fields:

- Agent Name:** A text input field.
- Type:** A dropdown menu currently set to "Coordinator".
- Agent Imports:** A table with a single header "Name" and an empty body.

At the bottom of the window, there are three buttons: "Add", "Edit", and "Delete".

# DESARROLLO DE SIGECO

Cuadro de diálogo para agregar/editar comportamientos:



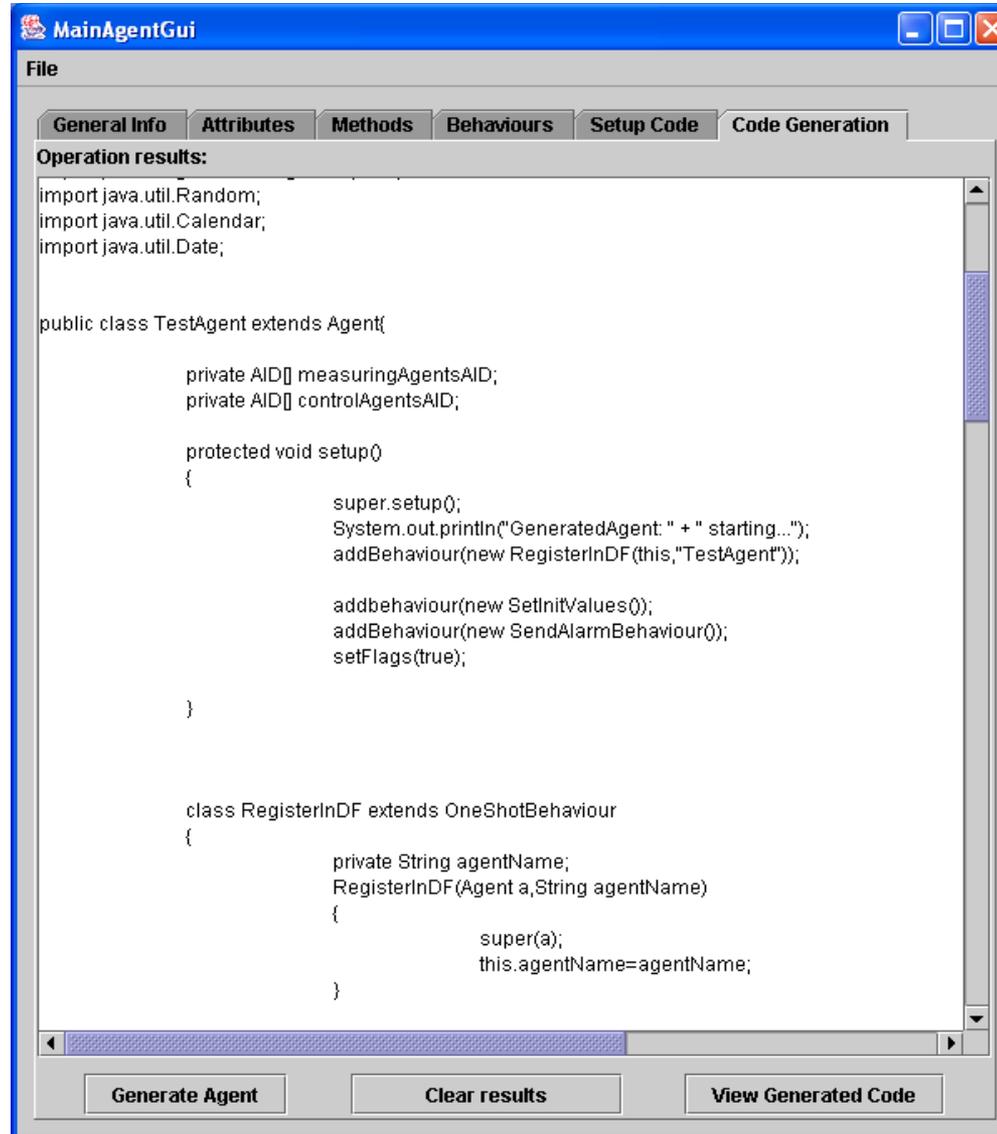
The image shows two overlapping dialog boxes. The larger one in the background is titled "Add a new behaviour" and has a blue header bar. It contains a "Name:" label followed by an empty text input field. Below this is a grey bar with the text "Please provide your behaviour code:". At the bottom of this dialog are three buttons: "Add", "Use wizard", and "Cancel".

The smaller dialog box in the foreground is titled "New Behaviour Wizard" and has a grey header bar. It contains the following elements:

- A label "Select a type of agent:" followed by a dropdown menu showing "Observation".
- A label "Select the type of behaviour that you want to add:" followed by a dropdown menu showing "measure".
- A label "Behaviour Name:" followed by a text input field containing "MeasureBehaviour".
- Two buttons at the bottom: "Add" and "Cancel".

# DESARROLLO DE SIGECO

Código  
fuente  
de un  
agente



The screenshot shows a window titled "MainAgentGui" with a menu bar containing "File". Below the menu bar are several tabs: "General Info", "Attributes", "Methods", "Behaviours", "Setup Code", and "Code Generation". The "Code Generation" tab is active, displaying the following Java code:

```
Operation results:
import java.util.Random;
import java.util.Calendar;
import java.util.Date;

public class TestAgent extends Agent{

    private AID[] measuringAgentsAID;
    private AID[] controlAgentsAID;

    protected void setup()
    {
        super.setup();
        System.out.println("GeneratedAgent: " + " starting...");
        addBehaviour(new RegisterInDF(this,"TestAgent"));

        addbehaviour(new SetInitValues());
        addBehaviour(new SendAlarmBehaviour());
        setFlags(true);
    }

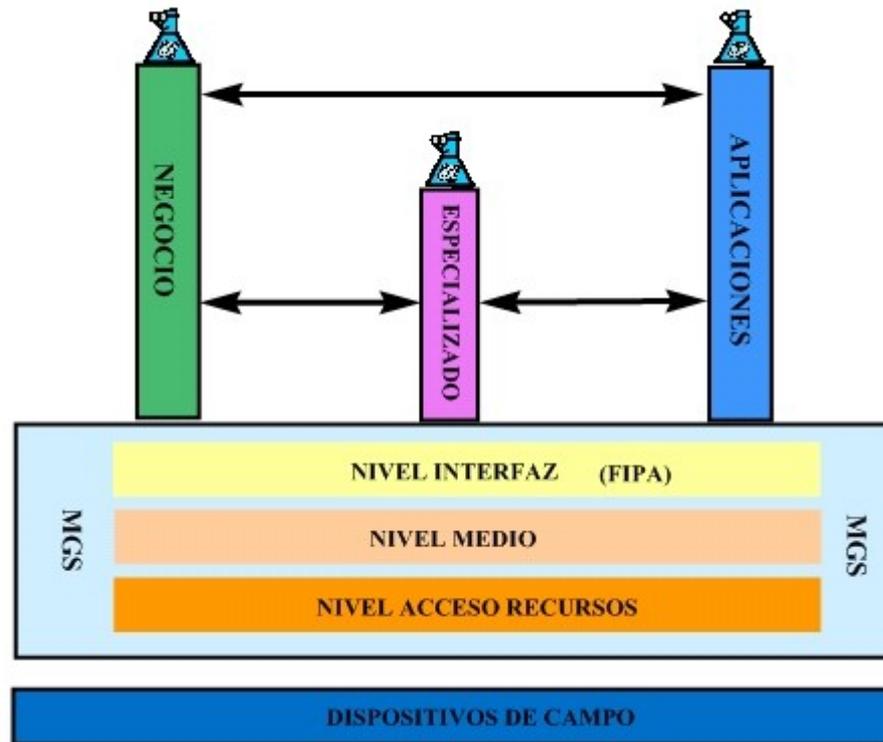
    class RegisterInDF extends OneShotBehaviour
    {
        private String agentName;
        RegisterInDF(Agent a,String agentName)
        {
            super(a);
            this.agentName=agentName;
        }
    }
}
```

At the bottom of the window, there are three buttons: "Generate Agent", "Clear results", and "View Generated Code".

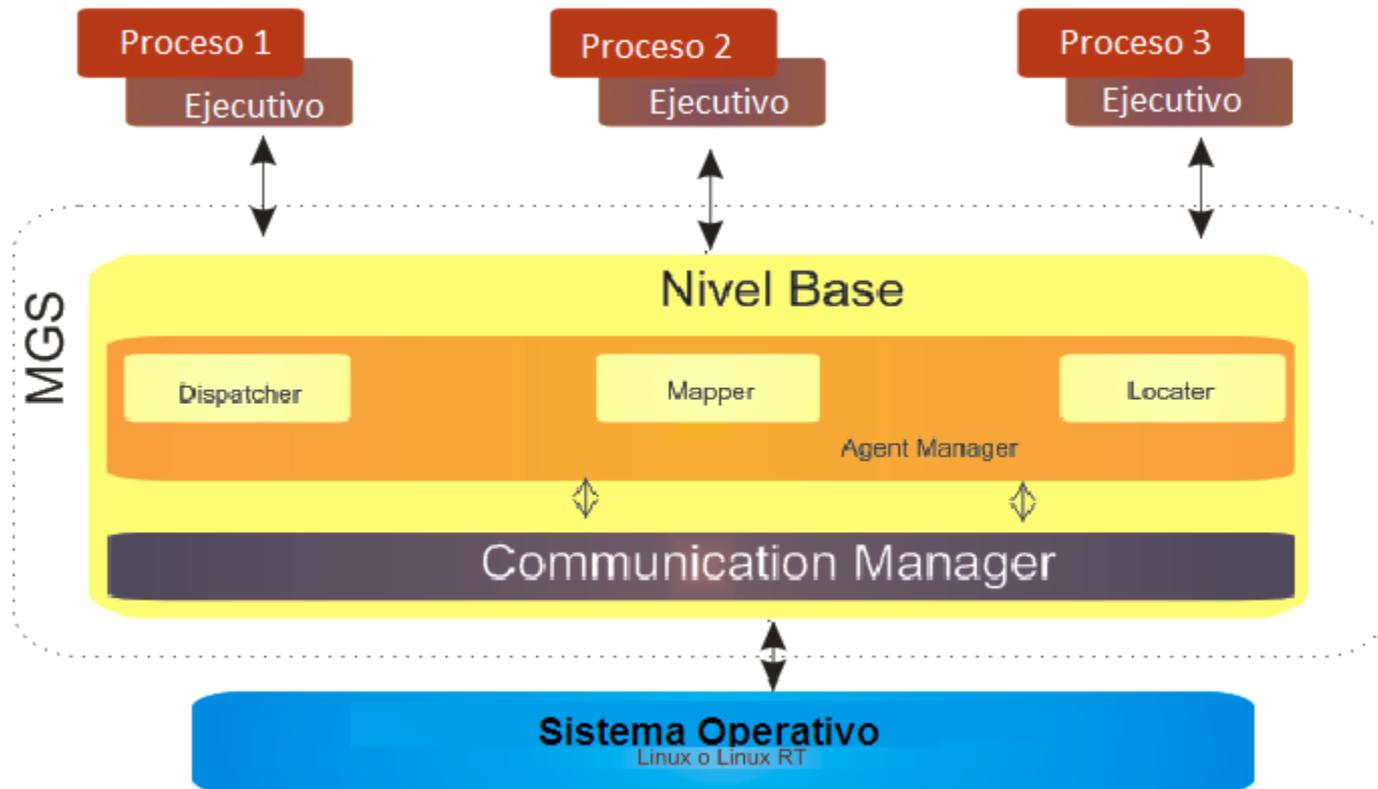


# Tiempo real en SMA

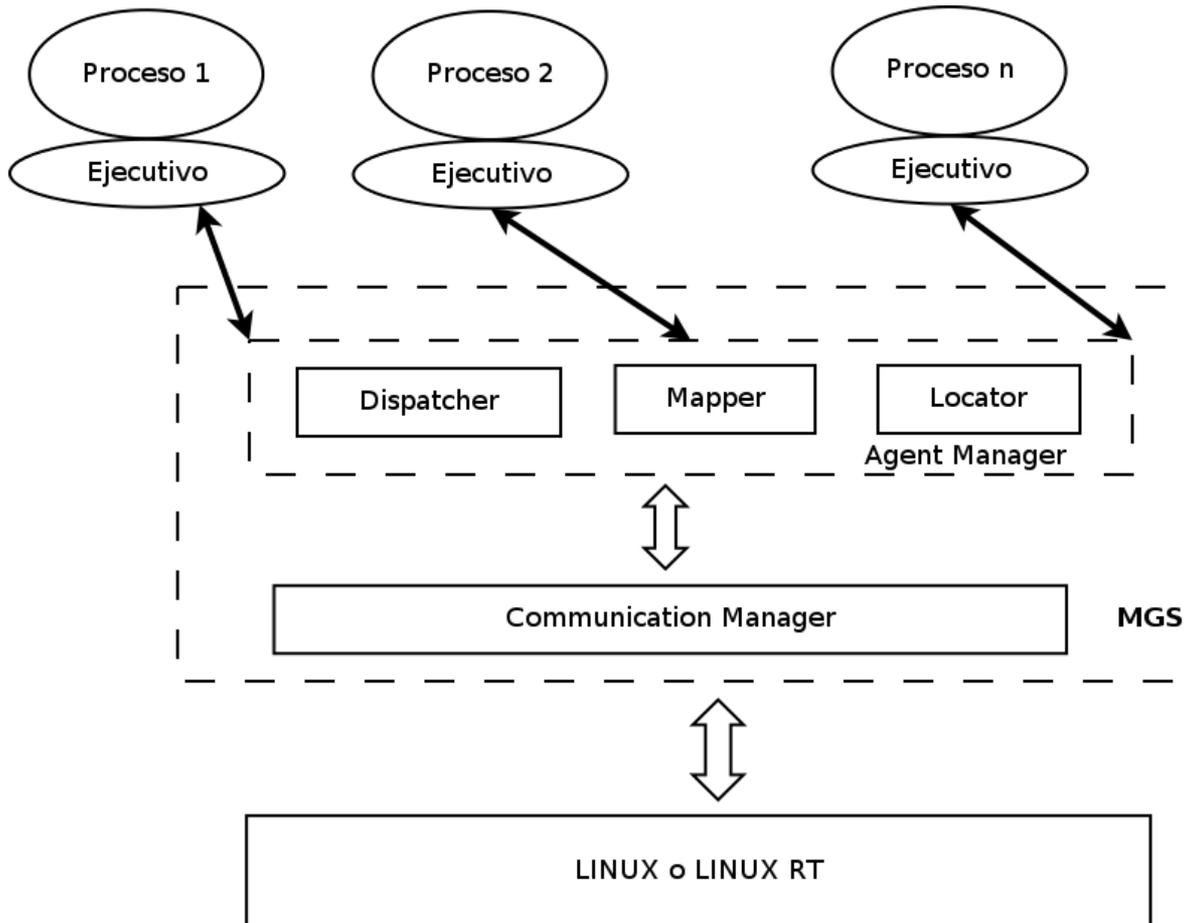
# MGS tiempo real



# MGS tiempo real

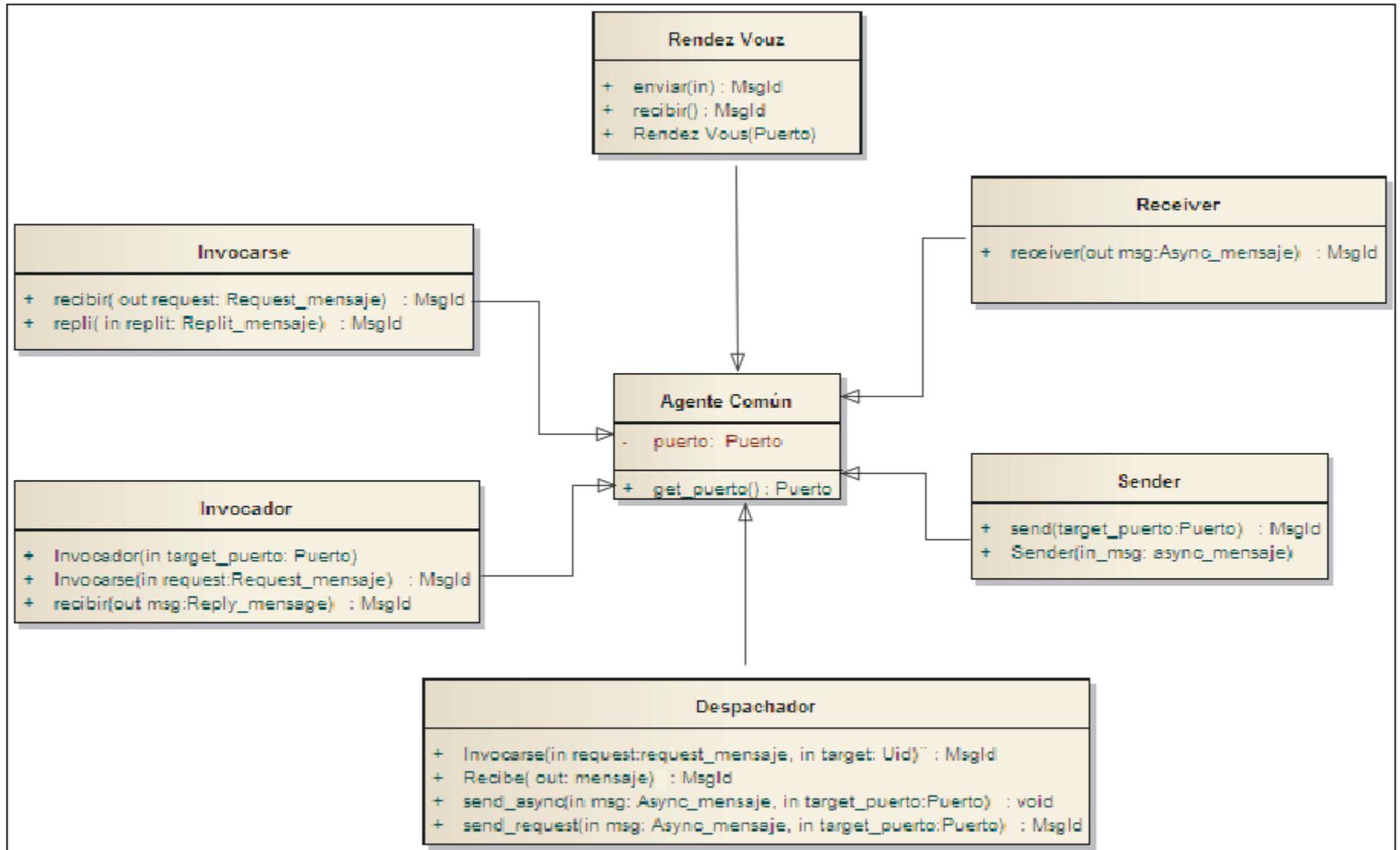


# MGS tiempo real

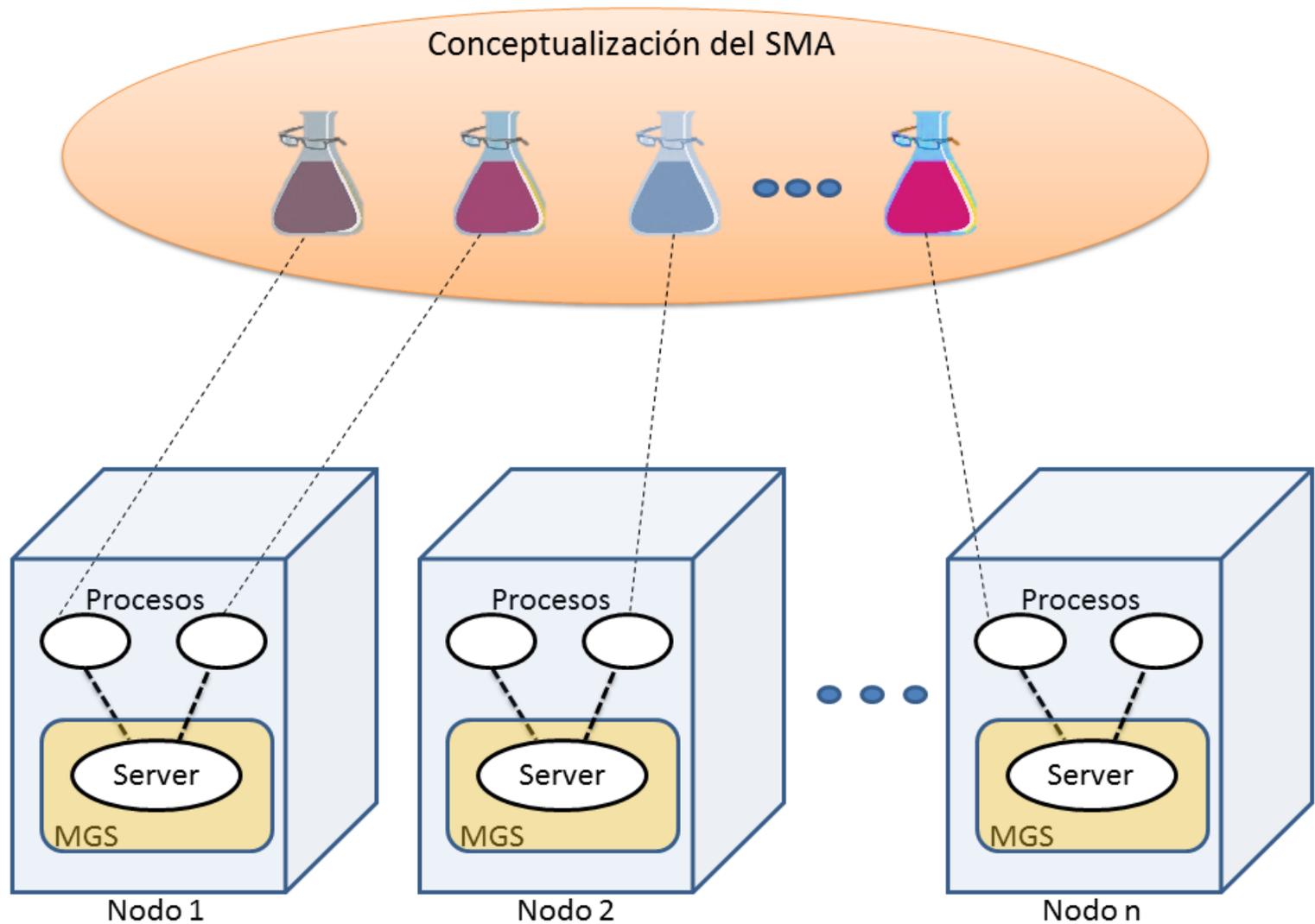


Nivel Medio

# Agente de Comunicación.



# Ambiente de ejecución distribuido.

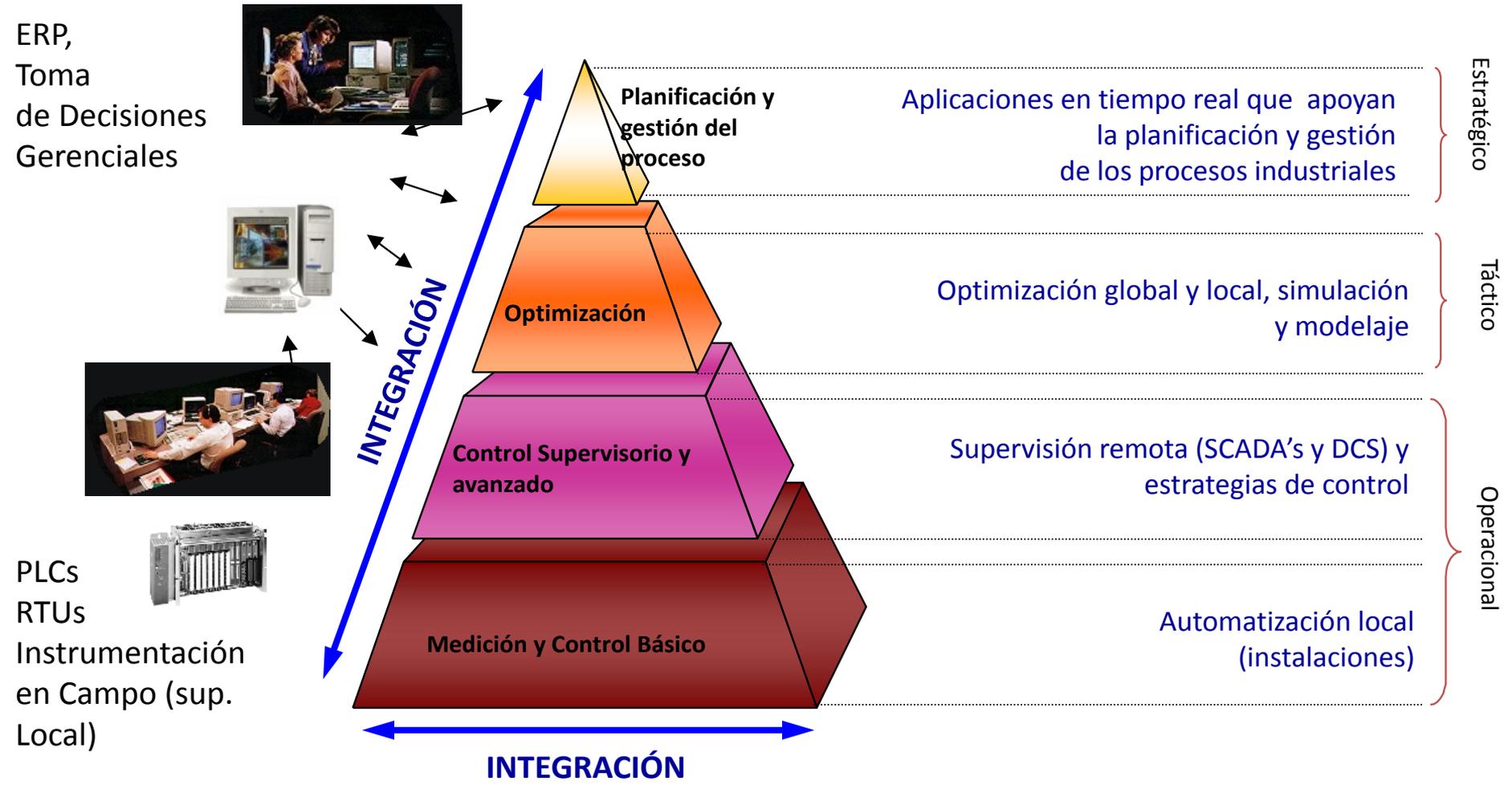


# Automatización Industrial y SMA

# Inteligencia Artificial Distribuida

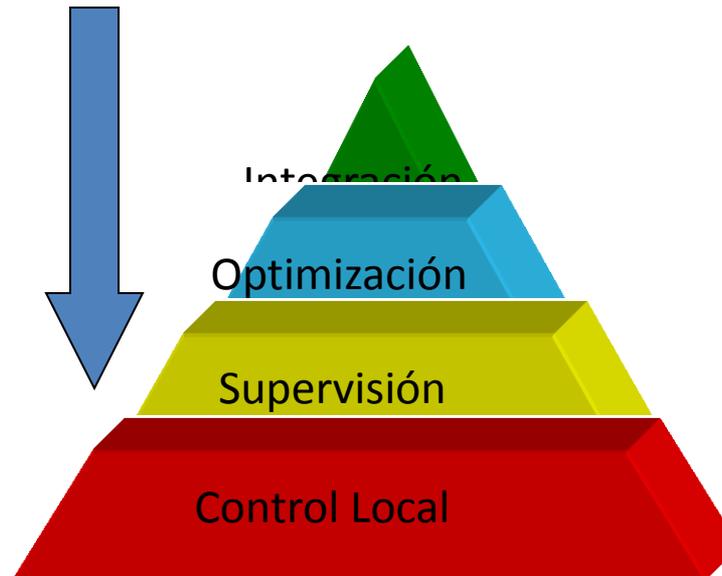
- ¿POR QUE PENSAR EN LA IAD EN LOS SCDI?:
  - LOS PROBLEMAS SON FISICA Y FUNCIONALMENTE DISTRIBUIDOS
  - LA COMPLEJIDAD DE LOS PROBLEMAS IMPONEN UNA VISION LOCAL Y DISTRIBUIDA A LA VEZ
  - LOS SISTEMAS DEBEN PODER ADAPTARSE A MODIFICACIONES DE SUS ESTRUCTURAS Y DEL AMBIENTE
  - LA INGENIERIA DE SOFTWARE VA HACIA UN DISEÑO EN TERMINO DE UNIDADES AUTONOMAS EN INTERACCION

# Automatización Industrial



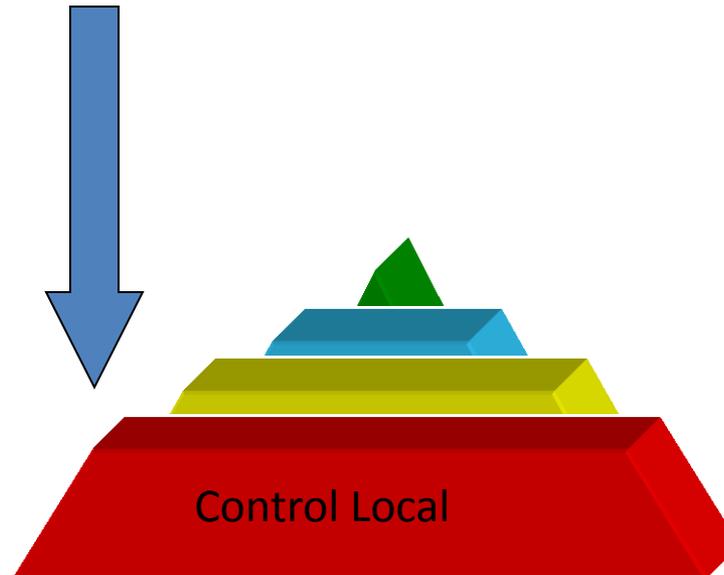
# Arquitecturas de AI

Aplanamiento  
de la Pirámide



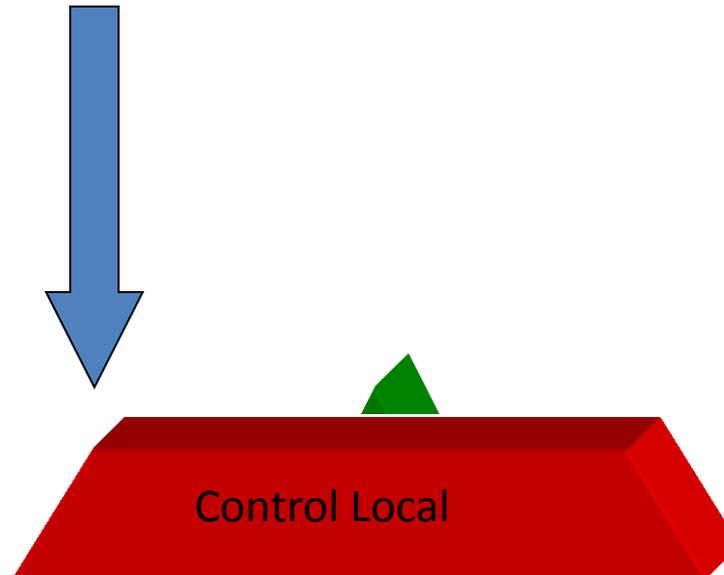
# Arquitecturas de AI

Aplanamiento  
de la Pirámide



# Arquitecturas de AI

Aplanamiento  
de la Pirámide



# Arquitecturas de AI

Aplanamiento de la Pirámide  
Distribución de la Inteligencia  
Autonomía de los Objetos de Negocio



**Plataforma de AI Distribuida**

¿Cómo Lograrlo?

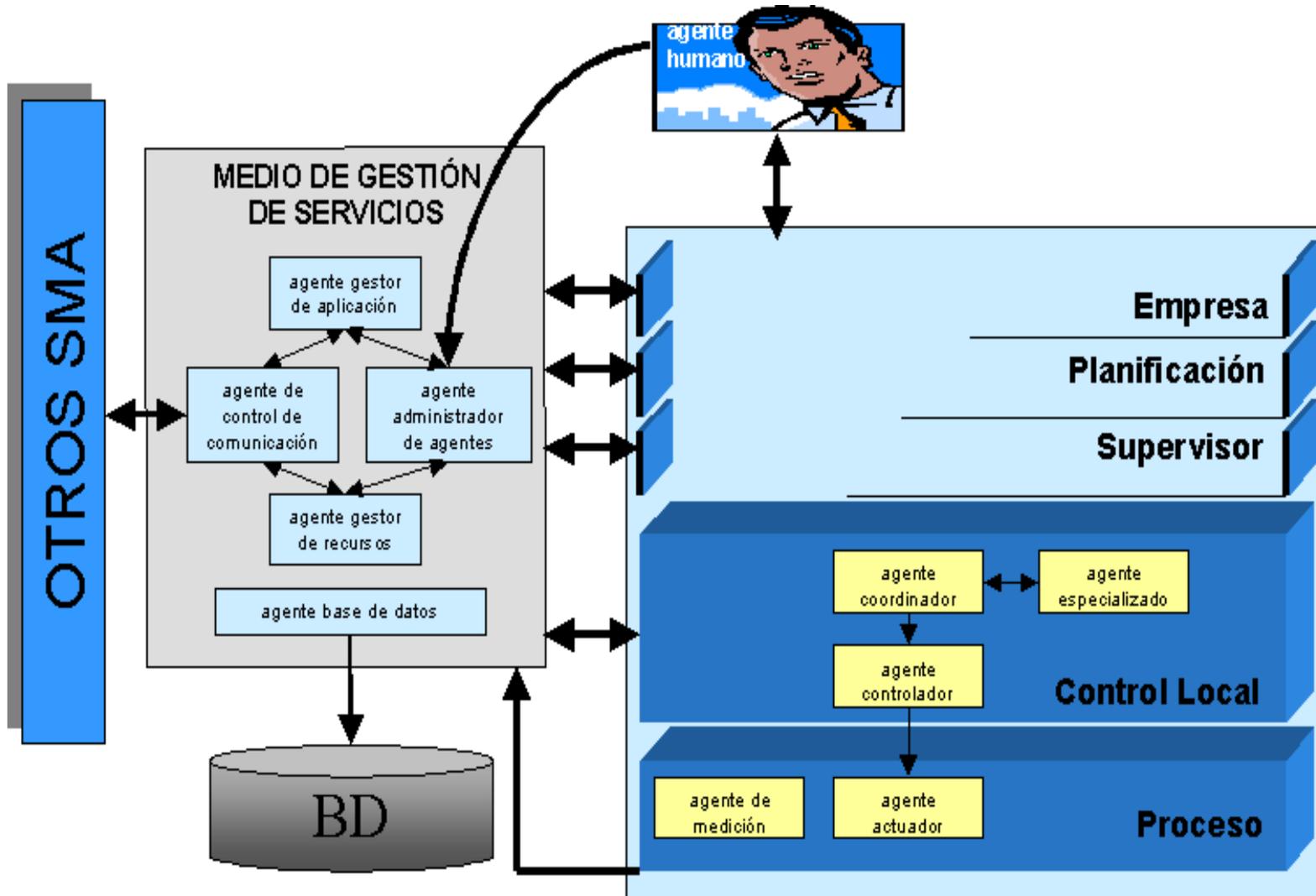
Nuevos Paradigmas de AI

# SCDIA

## Sistemas de Control Distribuido Inteligentes Basados en Agentes

- **Características:**
  - Sistema descentralizado más que centralizado,
  - Emergente más que planificado,
  - Autónomo más que controlado,
  - Concurrente más que secuencial.
- **Habilidades**
  - ➤ Aprendizaje
  - ➤ Búsqueda
  - ➤ Inferencia y predicción.
  - ➤ Representación de fuentes de conocimiento
  - ➤ Evolutivas.

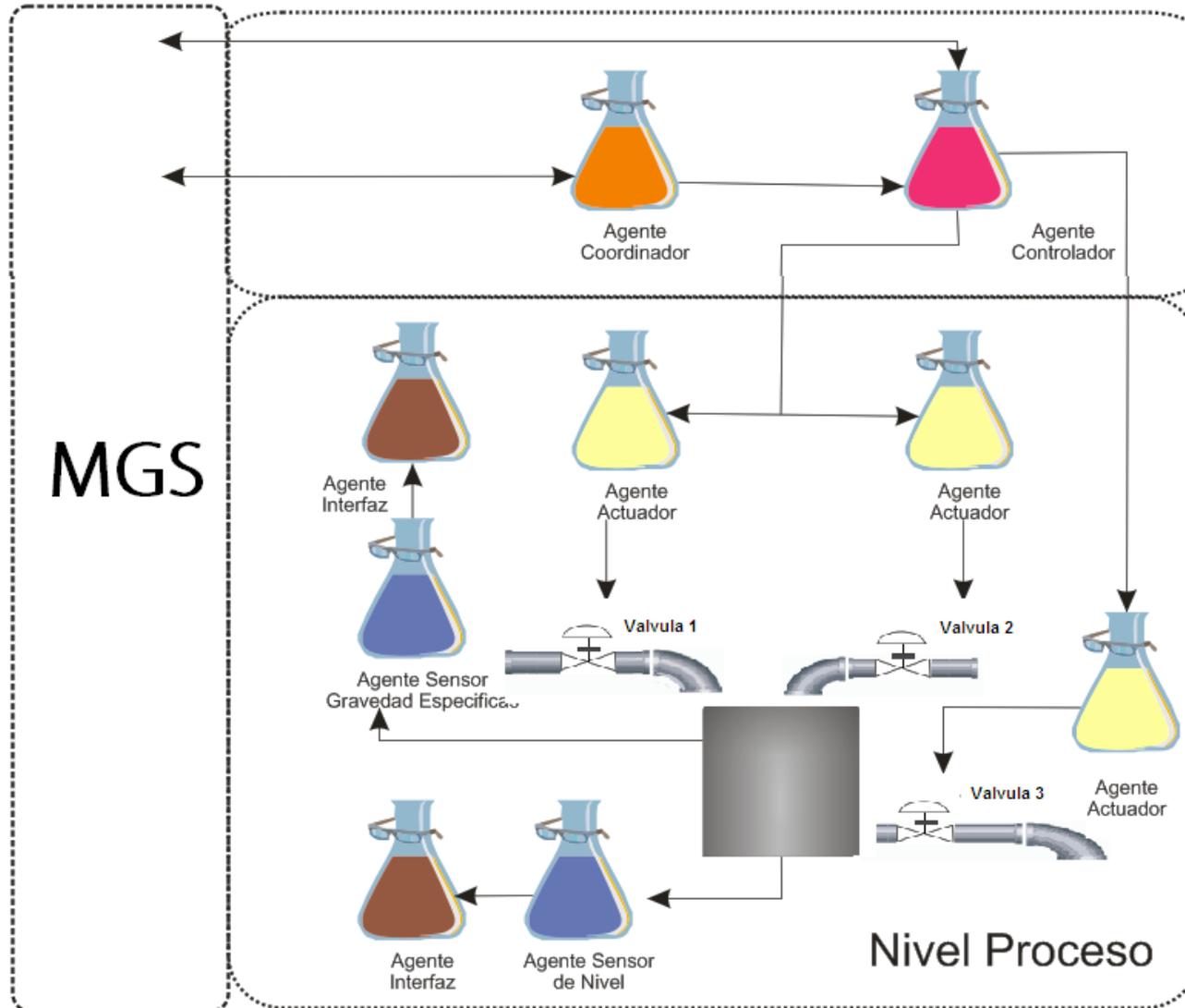
# SCDIA



# SCDIA

- Cinco agentes de control
  - Agente de observación/medición
  - Agente de Actuación
  - Agente Controlador
  - Agente Coordinador
  - Agente Especializado
- Sistema de Gestión de Servicios
  - Agente Administrador de Agentes
  - Agente Gestor de Recursos
  - Agente Gestor de Aplicaciones
  - Agente de Base de Datos
  - Agente de Control de Comunicaciones

# Sistema de control basado en el modelo SCDIA



# Tareas del SCDIA

## Tareas de Medición

- Medir – 1
  - Sensar – 1.1
  - Acondicionar – 1.2

- Procesar variable – 2

## Tareas de Control

- Toma de decisiones – 1
- Obtención acción de control – 2
- Procesar acción de control – 3
- Ejecutar acción de control – 4

## Tareas de especializado

- Procesar agente especializado - 1

## Tareas de manejo de información

- Almacenar información –1
- Actualizar información - 2
- Procesar (del A. BD) - 3
- Evaluar (del A. BD) - 4
- Buscar información - 5
- Gestión BD – 6

## Tareas de localización

- Localizar recurso
- Localizar agente especializado de recurso
- Localizar aplicación
- Loc. Ag. Especializado en aplicación
- Asignación de tarea/recurso
  - Optimización

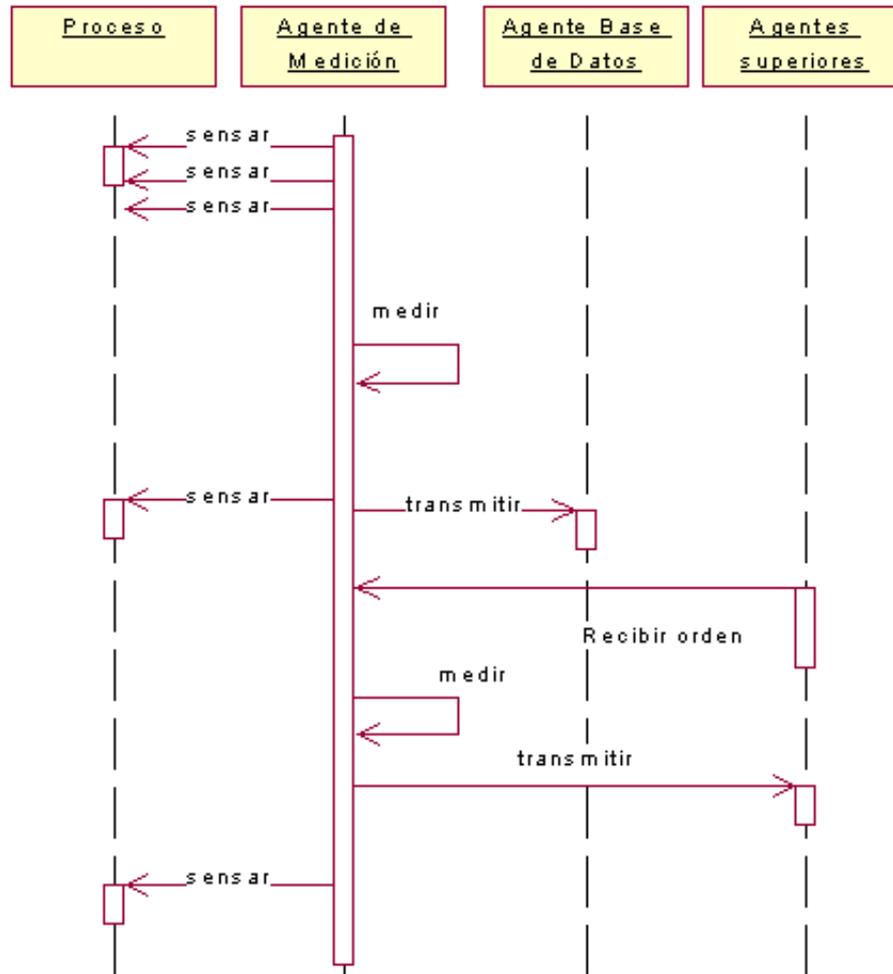
## Tareas de planificación

- Procesar Agente coordinador – 1
  - Evaluar
  - Tomar decisiones
  - Cambiar atributos de agentes

## Tareas de activación de agentes

- Inicializar agente – 1
  - Asociar nombre, nodo y estado – 1.1
  - Comunicar agente de recurso – 1.2
- Migrar agente – 2
- Cambiar estado – 3
- Administrar seguridad – 4
- Destruir agente – 5

# Diagrama de Interacción: Midiendo



# Ejemplo de Modelo de Inteligencia

## Agente Controlador

### *Mecanismo de Razonamiento*

<b>Fuente de Información:</b>	Agente de Observación, Agente Coordinador, SGS
<b>Fuente de Activación:</b>	Intervención
<b>Tipo de Inferencia:</b>	Deductiva, Inductiva, Absurdo
<b>Estrategia de Razonamiento:</b>	Sistema experto difuso, Redes Neuronales para reconocimiento de patrones y minería de datos. Algoritmos genéticos para búsqueda de soluciones.

### *Mecanismos de Aprendizaje*

<b>Nombre:</b>	Sistema Clasificador Difuso
<b>Tipo:</b>	Adaptativo
<b>Fuente de Aprendizaje:</b>	Información Histórica
<b>Mecanismo de Actualización:</b>	Modificación de reglas del motor de inferencia .

<b>Nombre:</b>	Máquinas de Aprendizaje
<b>Tipo:</b>	Adaptativo, en línea
<b>Fuente de Aprendizaje:</b>	Información histórica, Información de estado
<b>Mecanismo de Actualización:</b>	Modificación de la estructura interna de la máquina

# SADIA

- Modelar la arquitectura de automatización como SMA
- 3 niveles de abstracción
  - Objetos de Negocio
  - Actividades de Automatización
  - Funciones
- Cada nivel de abstracción conforma un SMA

# SADIA

## Primer Nivel de Abstracción

Objeto de Negocio 1

...

Objeto de Negocio n

## Segundo Nivel de Abstracción



Agente Ingeniería  
de Mantenimiento



Agente Manejo de  
Situaciones Anormales



Agente Control  
de Procesos



Agente Manejo de Factores  
de Producción



Agente Planificación de  
Producción

## Tercer Nivel de Abstracción

(Depende del Marco de Referencia a adoptar)



Agente 1

...

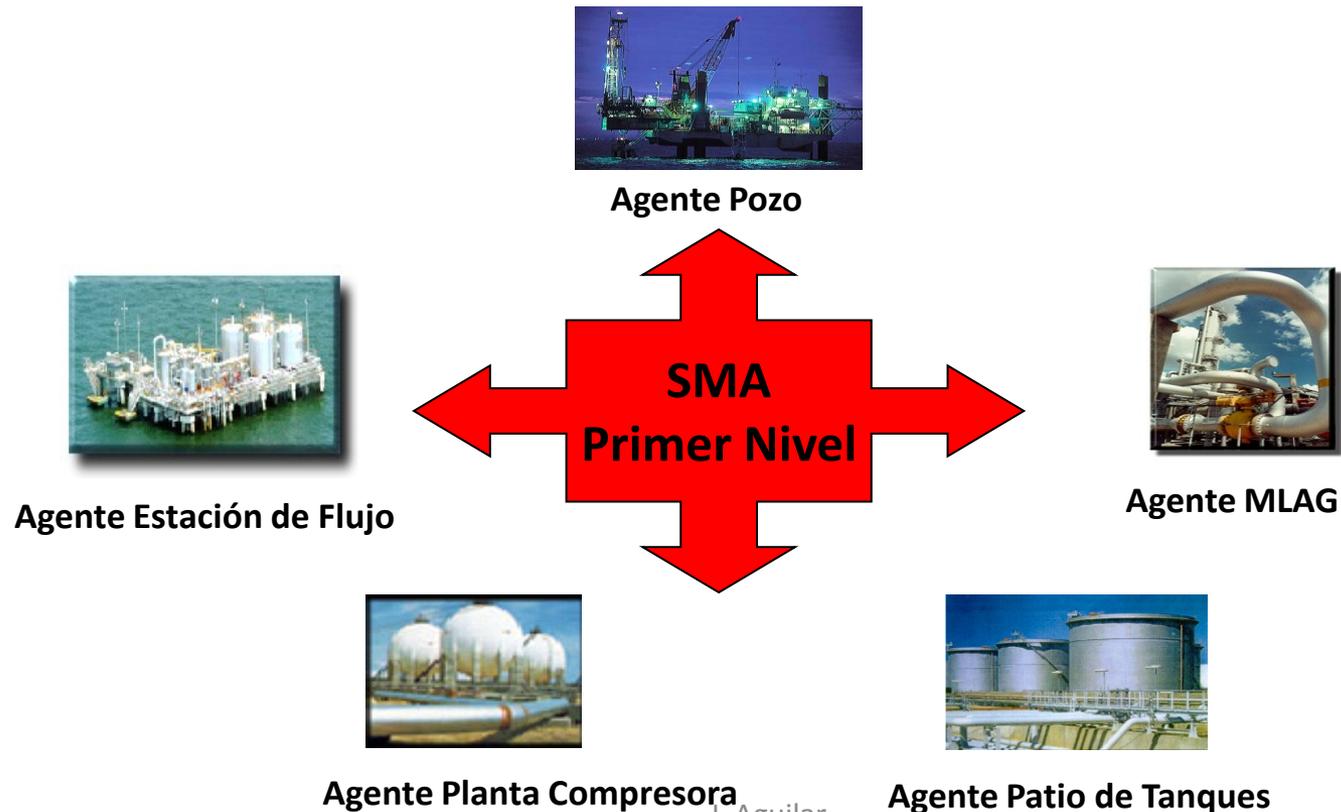


Agente n

# SADIA

## Primer Nivel de Abstracción

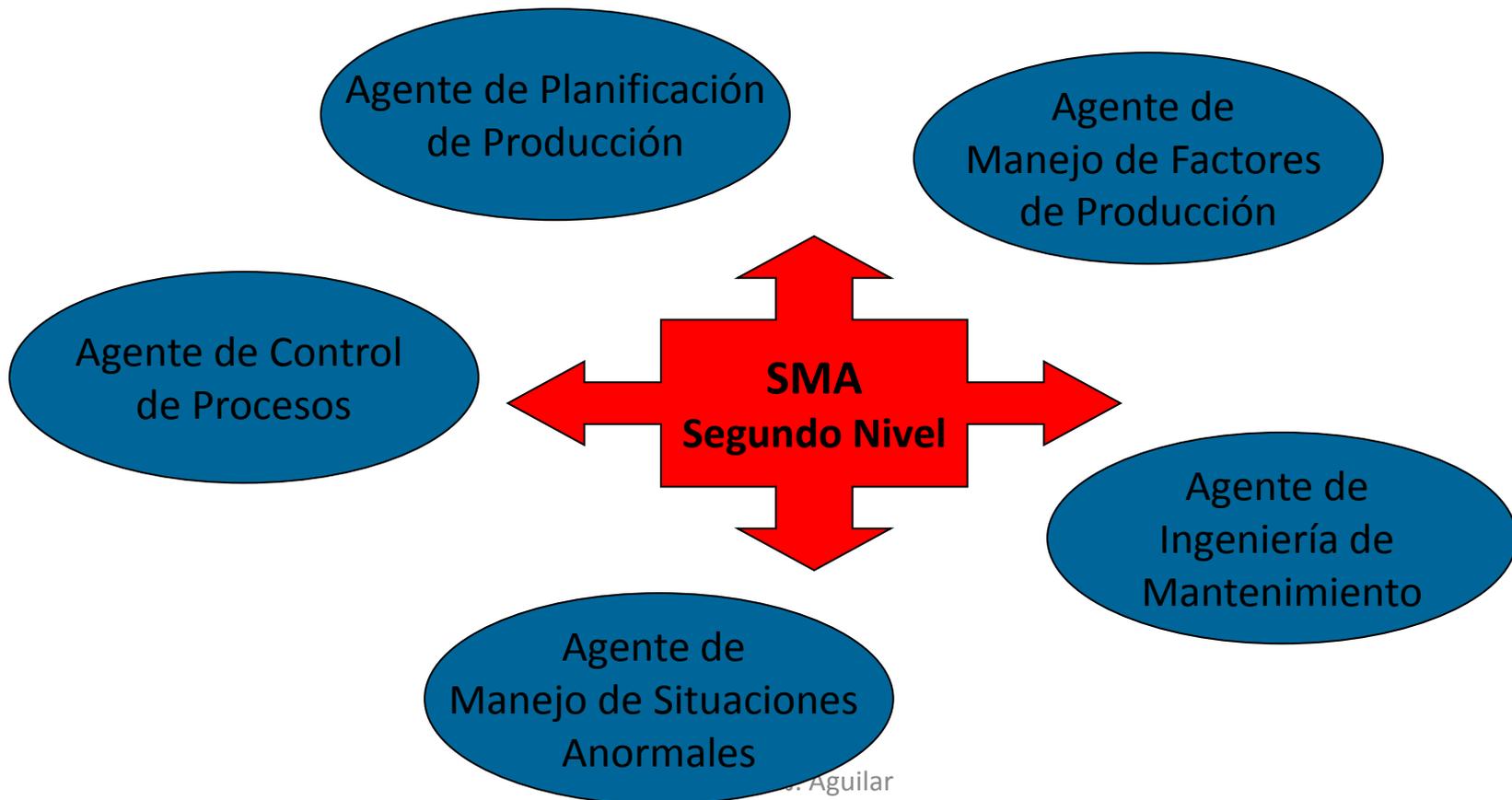
- Modela cada unidad de producción de la UP como un agente



# SADIA

## Segundo Nivel de Abstracción

- Modela las actividades vinculadas con la automatización industrial



# SADIA

## Tercer Nivel de Abstracción

### Agente del Segundo Nivel de Abstracción

Agente  
Coordinador

Agente  
Controlador

Agente  
Especializado

Agente  
de Observación

Agente  
de Actuación

Modela las funciones  
vinculadas con cada  
actividad del segundo  
nivel

Instrumentos  
Infraestructura  
Mantenimiento  
S.H.A.

**PROCESO**

SCADA  
Operadores  
Ing. Optimización  
Aplicaciones



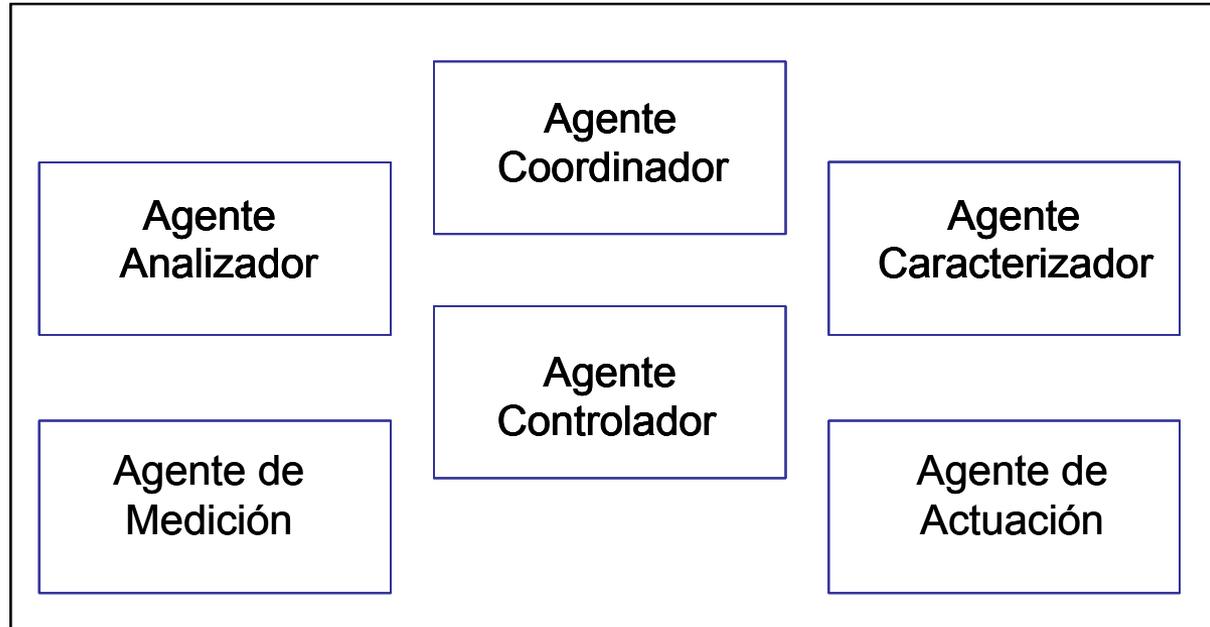
# Aplicaciones

## Agente Manejo de situaciones anormales

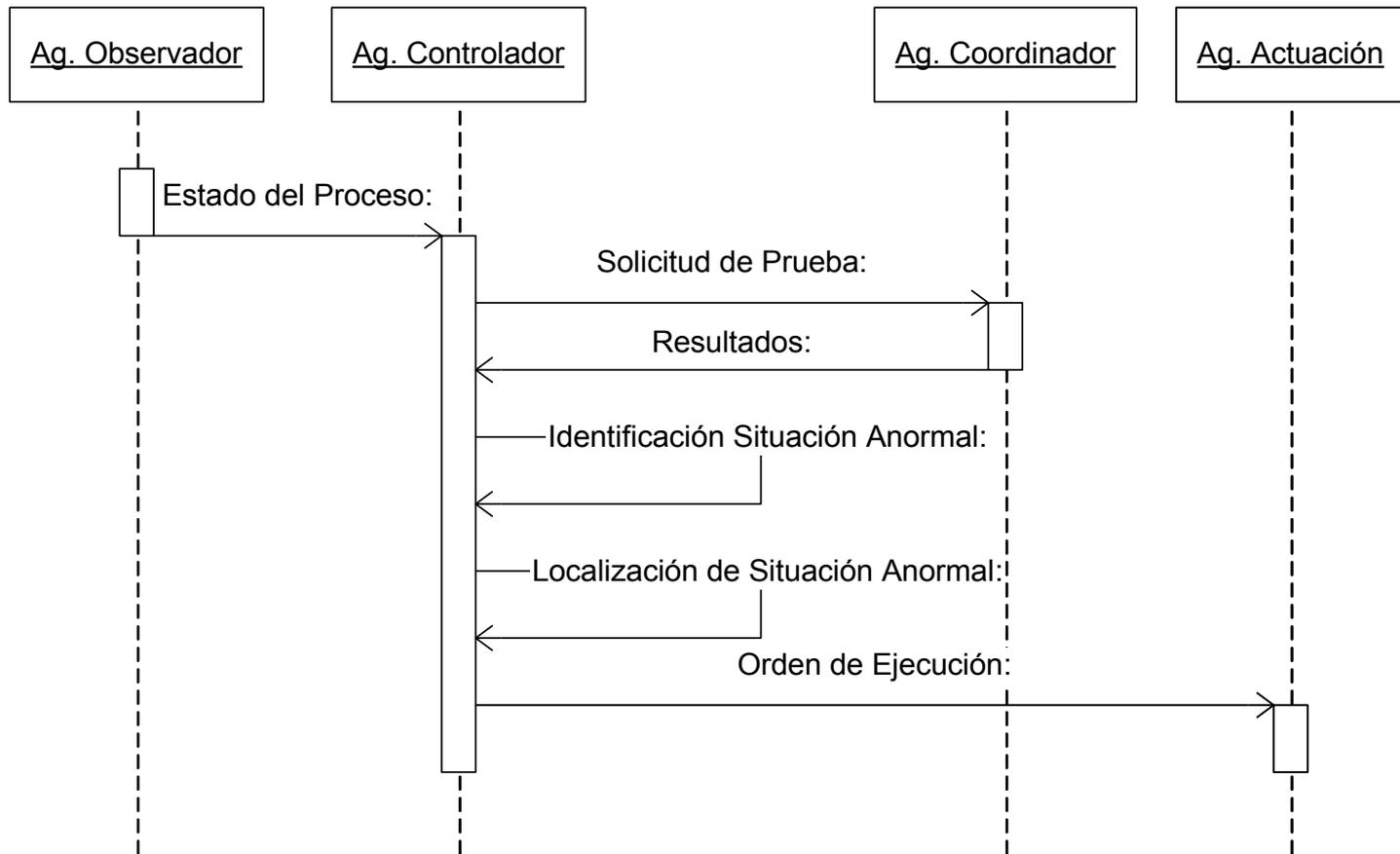
# Agente Manejo de situaciones anormales

- Una situación anormal es una condición, evento o circunstancia que involucra la pérdida de la capacidad de un elemento de realizar una función asociada a él, que hace que las operaciones se desvíen de su estado de operación normal
  - Las consecuencias pueden ser tanto mínimas como de alcances catastróficos.
  - Produce efectos (síntomas) que son las señales que permiten identificarla.
- **La meta** del manejo de situaciones anormales en la industria es mantener el desempeño operacional, mantener la disponibilidad continua de los activos de producción, minimizar los costos de operación, etc.
  - Para ello se deben detectar situaciones anormales en forma temprana y recomendar cursos de acción a seguir antes que ellas impacten la operación.

# AMSA



# AMSA



## Acto de Habla: Orden de ejecución

Nombre	Orden de ejecución
Tipo	Informar
Objetivo	Emitir orden de ejecución de una acción
Agentes	Controlador, actuación
Iniciador	Controlador
Precondición	Acción para ejecutar
Condición de terminación	Emisión de acción a ejecutar
Conversaciones	Obtención del control, ejecución de acciones
Descripción	El agente controlador emite una orden de ejecución de alguna acción al agente actuador

## Mecanismo de Razonamiento y de aprendizaje

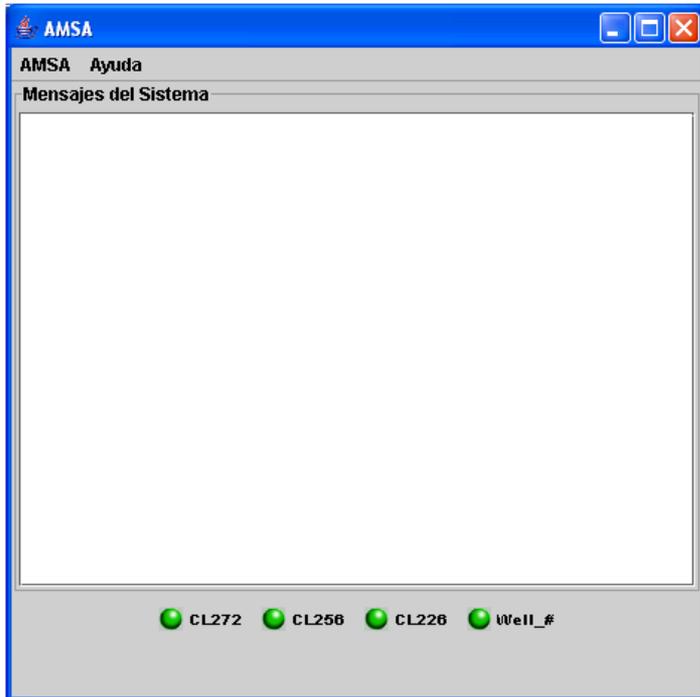
Fuente de Información	Agente de Observación, Agente Coordinador, SGS
Fuente de Activación	Intervención
Tipo de Inferencia	Deductiva
Estrategia de Razonamiento	Sistema experto

Tipo	Adaptativo
Técnica de Representación	Reglas
Fuente de Aprendizaje	Exitos o fracasos
Mecanismo de Actualización	Las experiencias previas son utilizadas para actualizar el conocimiento

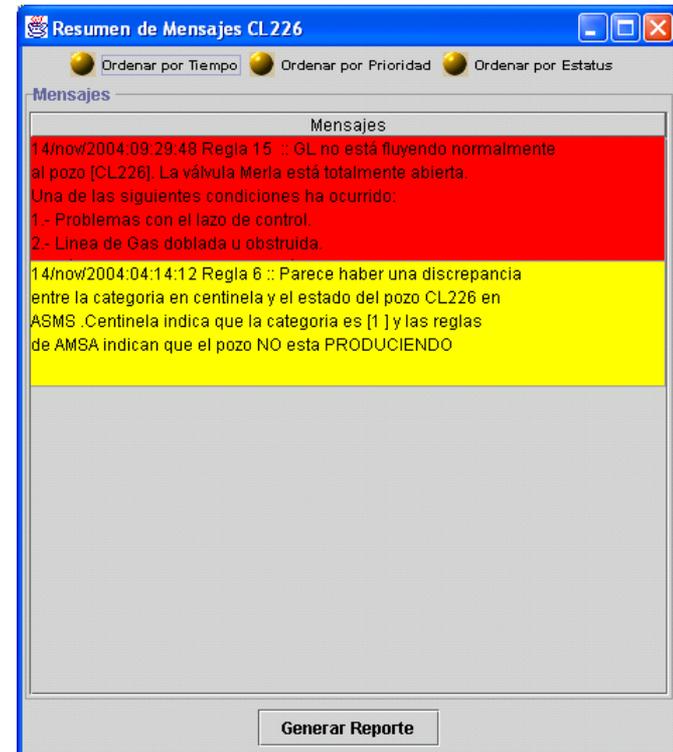
## *Reglas que forman el motor de inferencia del Agente Controlador*

Número de la Regla	Descripción de la Regla
1	Determina si un pozo esta produciendo, por medio del sensor de flujo.
2	Determina cuando hay una diferencia entre la señal Pozo_produciendo (obtenida de la regla 1) y la categoría del pozo almacenada en Centinela
3	Determina cuando ocurre una ruptura de la línea de producción mientras el pozo está en operación.
4	Determina cambios en el método de levantamiento artificial por comparación del método de diseño artificial con el método de operación.
5	Determina cuando a un pozo de GL se le está inyectando gas.
6	Determina cuando un pozo GL se cierra sin autorización.
7	Determina cuando el flujo de gas de levantamiento en un pozo está obstruido y, por lo tanto, el pozo no lo está recibiendo.
8	Determina cuando la línea que conecta el múltiple de GL y el pozo de GL sufre una ruptura mientras el pozo está recibiendo gas de GL.
9	Determina cuando la línea de producción de un pozo de GL está bloqueada u obstruida.
10	Determina una posible comunicación tubería-revestidor en pozos de GL continuos.
11	Determina una posible comunicación tubería-revestidor en pozos de GL continuos.
12	Determina cuando hay una comunicación mayor tubería-revestidor. Una comunicación mayor se define como aquella que causa una caída drástica en producción.

# AMSA



## Agente Actuador



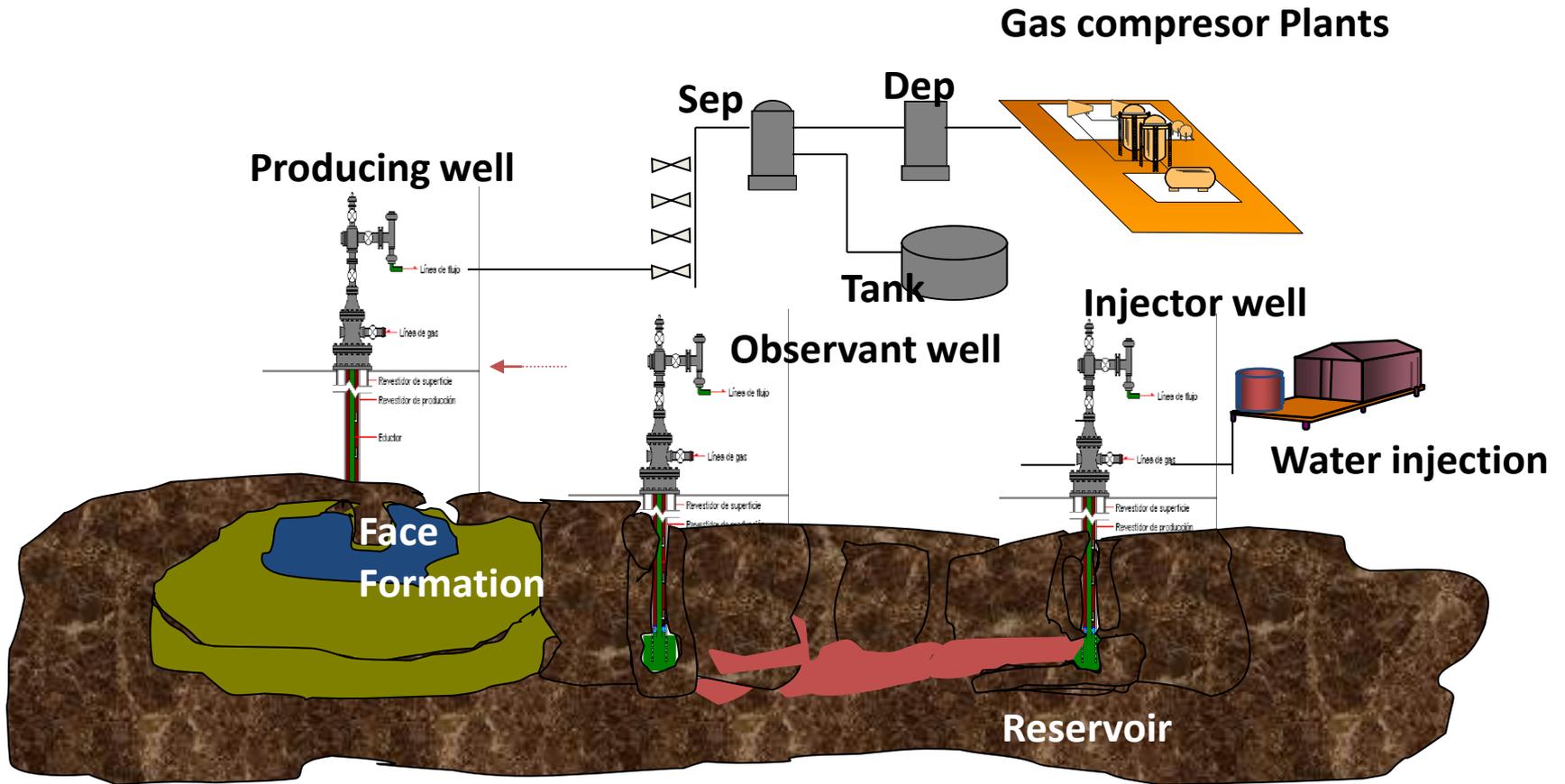
Los agentes Analizador, Caracterizador, Controlador, y Coordinador serán centralizados o distribuidos según lo requiera el sistema

Los agentes Observación y actuador están localizados en cada sitio

# **Ejemplo de Aplicación SADIA en un Proceso Industrial**

## **Caso Objeto de Negocio Agente Pozo**

# Campo Petrolero



**Tarea**

**Nombre:** Inyección Gas  
**Objetivo:** Inyectar Gas para permitir el fluido de liquido a la superficie  
**Descripción**  
**Pre-condición:** gas disponible, presión requerida

**Agente**

**Nombre:** Pozo GL  
**Posición:** BOL  
**Composicion de Agents:** según SADIA

**Objetivo**

**Nombre:** Extracción de aceite  
**parámetros:** entrada Metas de la producción  
**parámetros:** salida Producción actual, activación  
**Cond de la evaluación de funcionamiento.:** agente llevado Extremo **Cond.:** agente matado **Éxito**  
**Cond.:** Metas de la producción alcanzadas **Fall**  
**Cond.:** Metas de la producción unreached  
**Ontología:** Ontología de la automatización

**Servicio**

**Nombre:** Optimización de la elevación de gas  
**Parámetro entrada:** gas disponible, información del infraestructura,  
**parámetros salida:** elevación de gas, técnicas de optimización

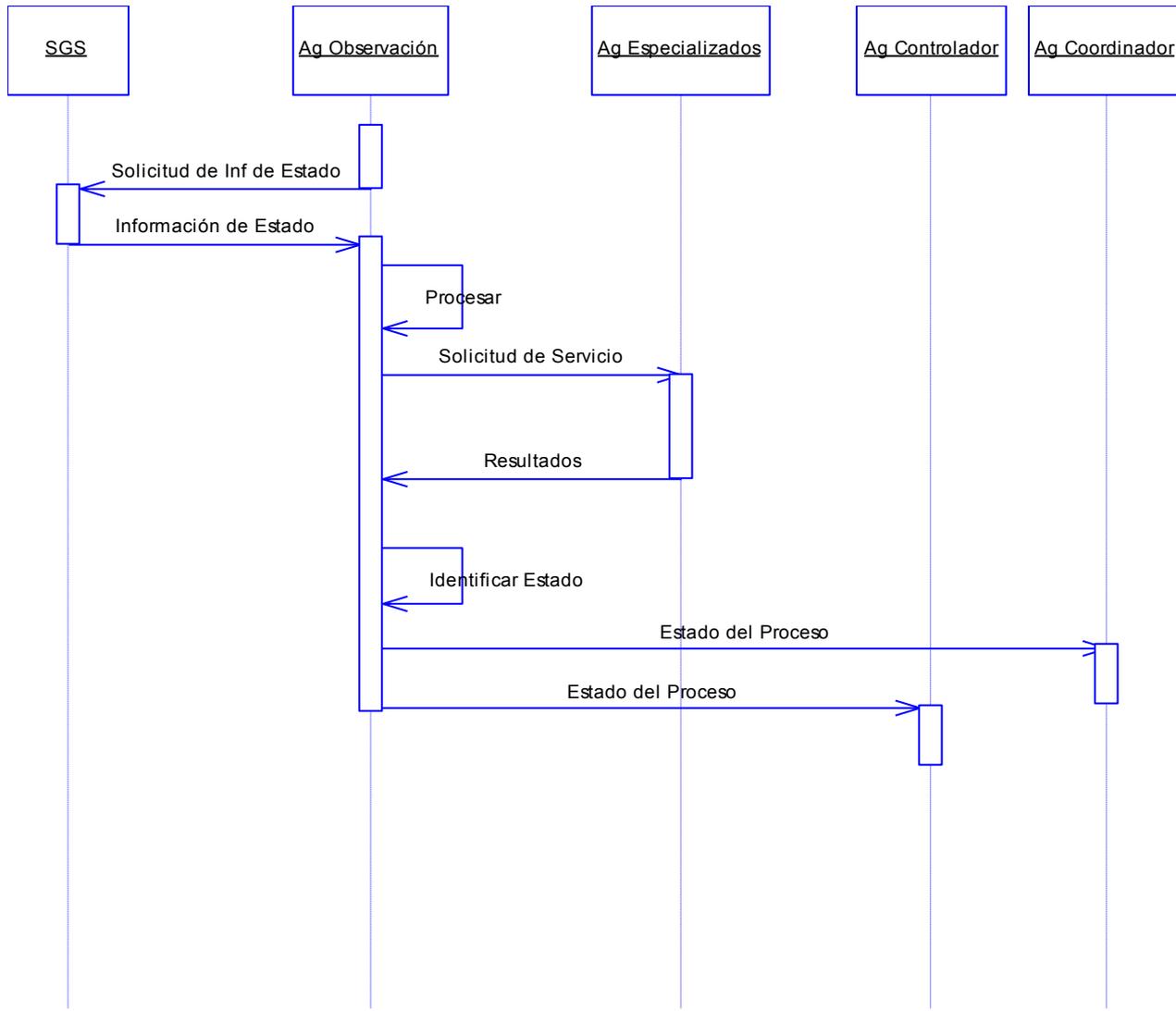
# Agente Pozo

## Modelo de Tareas Agente AMSA (Agente 2do. Nivel)

Agente	Tareas
<b>Agente de Manejo de Situaciones Anormales</b>	
Agente de Observación	Obtener información de estado
	Pedir datos ABD
	Procesar datos provenientes del proceso
	Caracterización de Pozos
	Identificación de situación anormal
	Emisión de alarmas
	Emisión de mensajes
Agente Controlador	Detección de síntoma
	Diagnóstico
	Identificación de causas
	Identificación de consecuencias
	Evaluación de criticidad
	Evaluación del proceso
Emitir orden de ejecución	

# Agente Pozo: (1er Nivel)

## Agente AMSA (2do Nivel)





# Agente Pozo: (1er Nivel)

## Agente AMSA (2do Nivel)

### Agente Controlador (3er Nivel)

#### Agente Controlador

##### *Mecanismo de Razonamiento*

- Fuente de Información:** Agente de Observación, Agente Coordinador, SGS
- Fuente de Activación:** Intervención
- Tipo de Inferencia:** Deductiva, Inductiva, Absurdo
- Estrategia de Razonamiento:** Sistema experto difuso, Redes Neuronales para reconocimiento de patrones y minería de datos. Algoritmos genéticos para búsqueda de soluciones.

##### *Mecanismos de Aprendizaje*

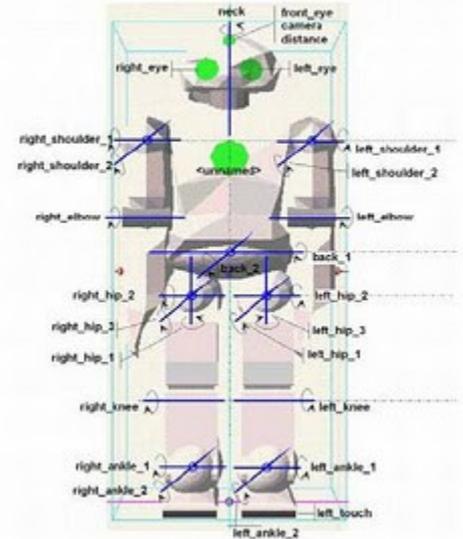
- Nombre:** Sistema Clasificador Difuso
- Tipo:** Adaptativo
- Fuente de Aprendizaje:** Información Histórica
- Mecanismo de Actualización:** Modificación de reglas del motor de inferencia .
- 
- Nombre:** Máquinas de Aprendizaje
- Tipo:** Adaptativo, en línea
- Fuente de Aprendizaje:** Información histórica, Información de estado
- Mecanismo de Actualización:** Modificación de la estructura interna de la máquina

# Aplicaciones

# Robótica

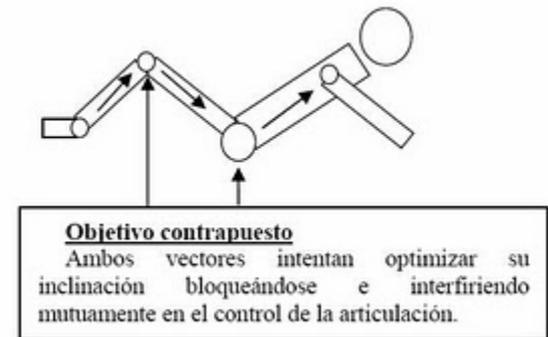
# Robots como un Sistema Multiagentes

- Coordinar los movimientos de un robot con múltiples grados de libertad es un problema que se complica a medida que el número de articulaciones y la complejidad de sus maniobras crecen.
- El problema de realizar una maniobra en robots complejos puede enfocarse como un problema de coordinación multi-objetivo.
- Esto da lugar a un problema clásico de optimización multiobjetivo con objetivos contrapuestos, es decir, la optimalidad de algunas inclinaciones de partes del cuerpo se contraponen a la optimalidad de otras durante algunos intervalos de tiempo,



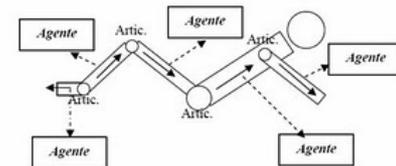
# Robots como un Sistema Multiagentes

- Se puede plantear como un problema de coordinación multiagente donde cada parte del cuerpo es un agente que intenta maximizar alguna función local para optimizar una función global y compite por uno o más recursos que son las articulaciones que están directamente relacionadas con él.
- Entonces el problema es la optimización de una función global mediante la coordinación de un sistema multiagente:  
*cada agente tiene que colaborar para lograr dicho objetivo global.*



# Robots como un Sistema Multiagentes

- Cada parte del cuerpo del robot tiene un vector de orientación asociado. Y el objetivo global se define, entre otras cosas, por una orientación dada en los vectores del robot. Así tendremos un agente por cada uno de los vectores del robot.
- Tendríamos dos opciones:
  - Cada agente actúa solo sobre sus articulaciones adyacentes intentando maximizar el objetivo global.
  - Cada agente actúa solo sobre sus articulaciones adyacentes intentando maximizar su objetivo local.
- De esta forma, sobre cada articulación competirían uno o dos agentes, haciendo más sencilla la política de asignación de recursos.



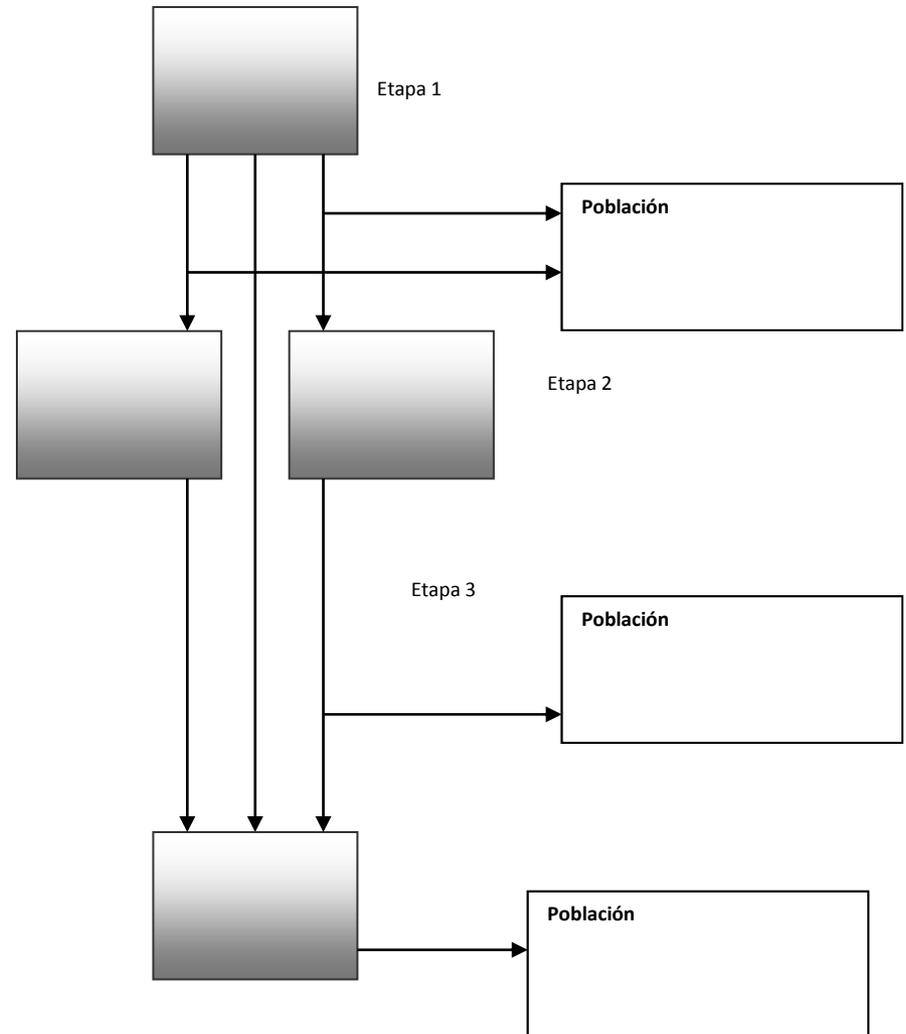


# Aplicaciones

## Sistema de Distribución de Agua

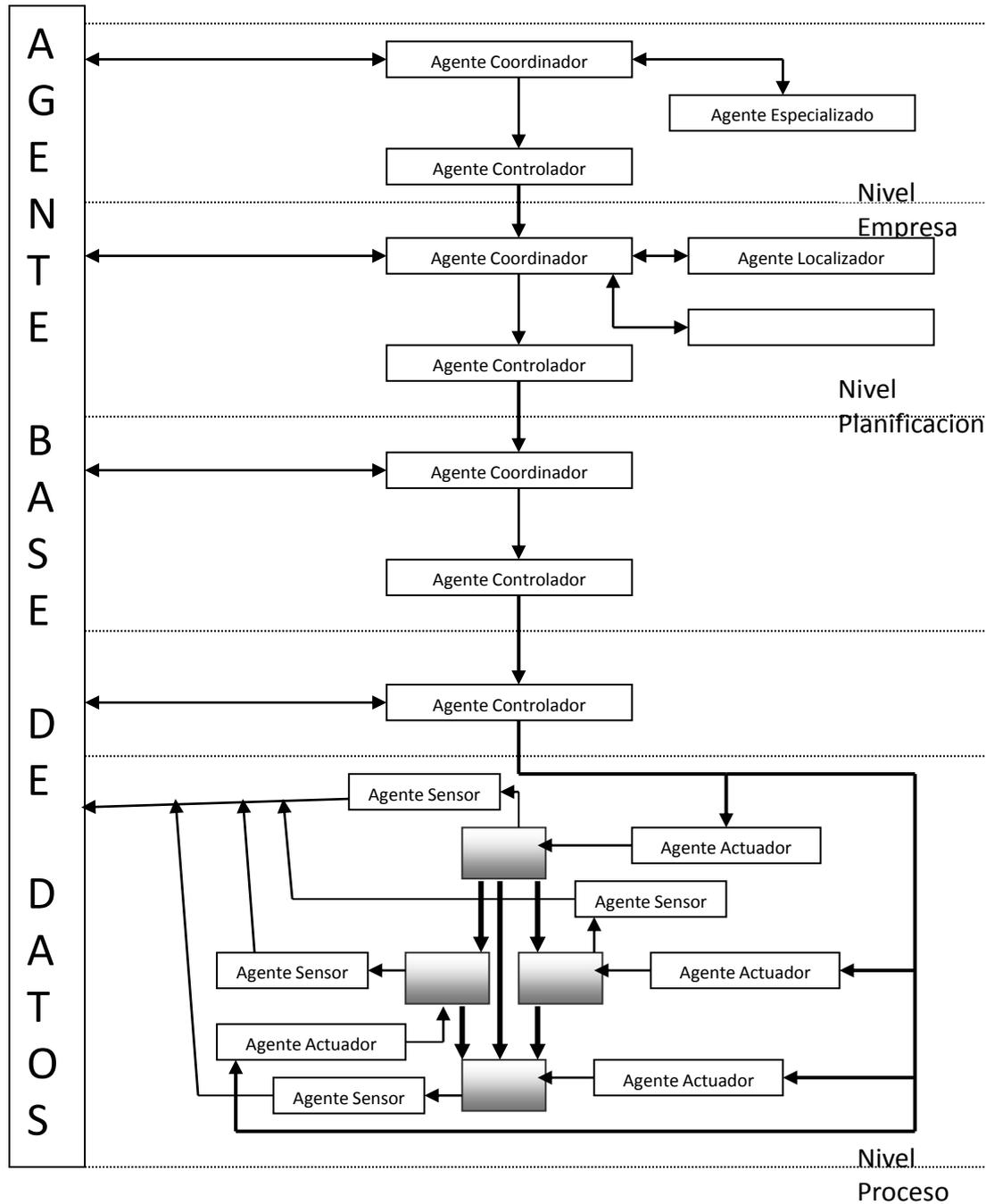
# Sistema de Distribución de Agua

- **Objetivo:**  
Ofrecer una distribución de agua que cumpla con las necesidades de toda la población.
- **Objetivo de Control:**  
Ajustar la velocidad de las bombas  
Para garantizar un flujo óptimo de agua,  
manteniendo los niveles de los tanques  
según los requerimientos del propio sistema.



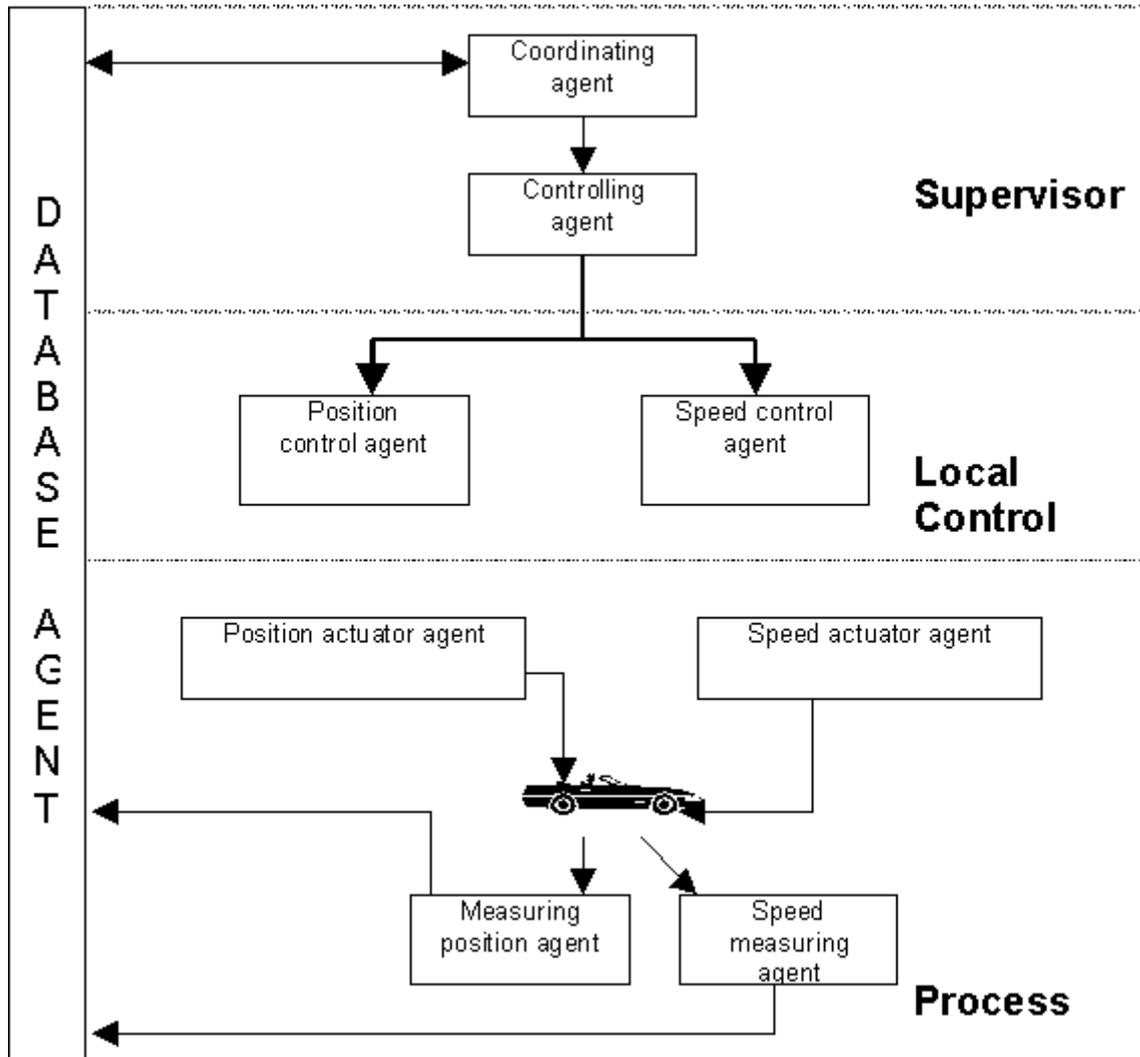


UNIVERSIDAD  
DE LOS ANDES  
MÉRIDA VENEZUELA



# Control de un Vehículo

# Control de un Vehículo



### Automatización Industrial basada en Agentes Inteligentes

Desde su inicio, en septiembre de 1986, el trabajo y entusiasmo puesto por profesores y generaciones de estudiantes han marcado una senda visible de resultados desarrollados en el marco del Programa de Maestría en Ingeniería de Control y Automatización de la Universidad de Los Andes. Numerosas publicaciones científicas, en medios de reconocido prestigio internacional, han permitido avalar ese esfuerzo, que ha abarcado áreas de aplicaciones como el Control de Procesos, el Diseño de Sistemas No Lineales, la Automatización Industrial y la Confiabilidad Operacional, entre otras. Esta obra, realizada por el Ing. Wilber Acuña, bajo la supervisión académica del Prof. Addison Ríos, en el terreno específico de los sistemas tolerantes a fallas, es también parte de ese mismo esfuerzo colectivo; iniciado por la determinación sin tregua y con visión de futuro del Prof. Ferenc Szigeti, maestro de generaciones de cuadros científicos, por lo cual esta obra es un homenaje sincero a su indiscutible labor en función de la ciencia y de la formación de excelentes especialistas.

El libro, que está dirigido a estudiantes de postgrado, académicos y especialistas, inicia con una exposición general sobre las fallas en sistemas dinámicos, para luego abordar el tema de la síntesis de filtros de detección robusta de fallas y el problema de separación de las mismas. Seguidamente realiza una exposición sobre estrategias anti-windup basada en el muy particular enfoque de las desigualdades lineales matriciales y haciendo uso de un exquisito dominio de las matemáticas. Finalmente, el libro incluye el diseño de filtros de detección de saturaciones, extendiendo las ideas de la generación de residuos provenientes de fallas para llevar a cabo la implementación de una estrategia de control acotado denominada compensación anti-windup

*Mérida, Abril 2011 Addison Ríos*

Automatización Industrial basada en Agentes Inteligentes

## Automatización Industrial basada en Agentes Inteligentes

