Curso IAD A-2012

Mi nombre: Jose Aguilar

Sitio de Trabajo: CEMISID

Contacto: aguilar@ula.ve

Consulta: por email cuadrar cita (en principio lunes en la tarde y miercoles en la mañana)

http://www.ing.ula.ve/~aguilar/

Enviarme correo (titulo mensaje estudiante <curso>)

OBJETIVO

Introducción a los sistemas multiagentes.

- Se presentan los diferentes problemas al querer diseñar sistemas multiagentes
- Se hace especial hincapié en los problemas de interacción
- Se estudian aspectos recientes vinculados a las propiedades emergentes en los sistemas multiagentes
- Se presenten algunos modelos de referencias

Conocimiento de base

- **Unidad I: Introducción**
  - Tema 1. Ideas de base de la IA
  - Tema 2. SMA y RPD.
  - Tema 3. Metodologías de desarrollo de SMA

- **Unidad II: Principio de los SMA**
  - Tema 1. Principio de los SMA: organización, características y estructura.
  - Tema 2: Problema de Confianza y Reputación
  - Tema 3. Interacción y sus tipos en los SMA

Conocimiento de base

- **Unidad III: Interacción y Comunicación en los SMA**
  - Tema 1: Cooperación,
  - Tema 2. Coordinación,
  - Tema 3: Negociación.
  - Tema .4 Comunicación: Lenguajes, actos de habla, Ontologías

- **Unidad IV: Aspectos Avanzados**
  - Tema 1: Aprendizaje en los SMA
  - Tema 2: SMA auto-organizados e Inteligencia Colectiva
  - Tema 3. Arquitecturas de Referencia en los SMA: Modelos de Referencia (FIPA, SCDIA, etc.), Estándares y Plataformas

BIBLIOGRAFIA

J. Aguilar, M. Cerrada, F. Hidrobo, A. íos, “SMA y sus apliaciones en Automatización Industrial”, Por salir

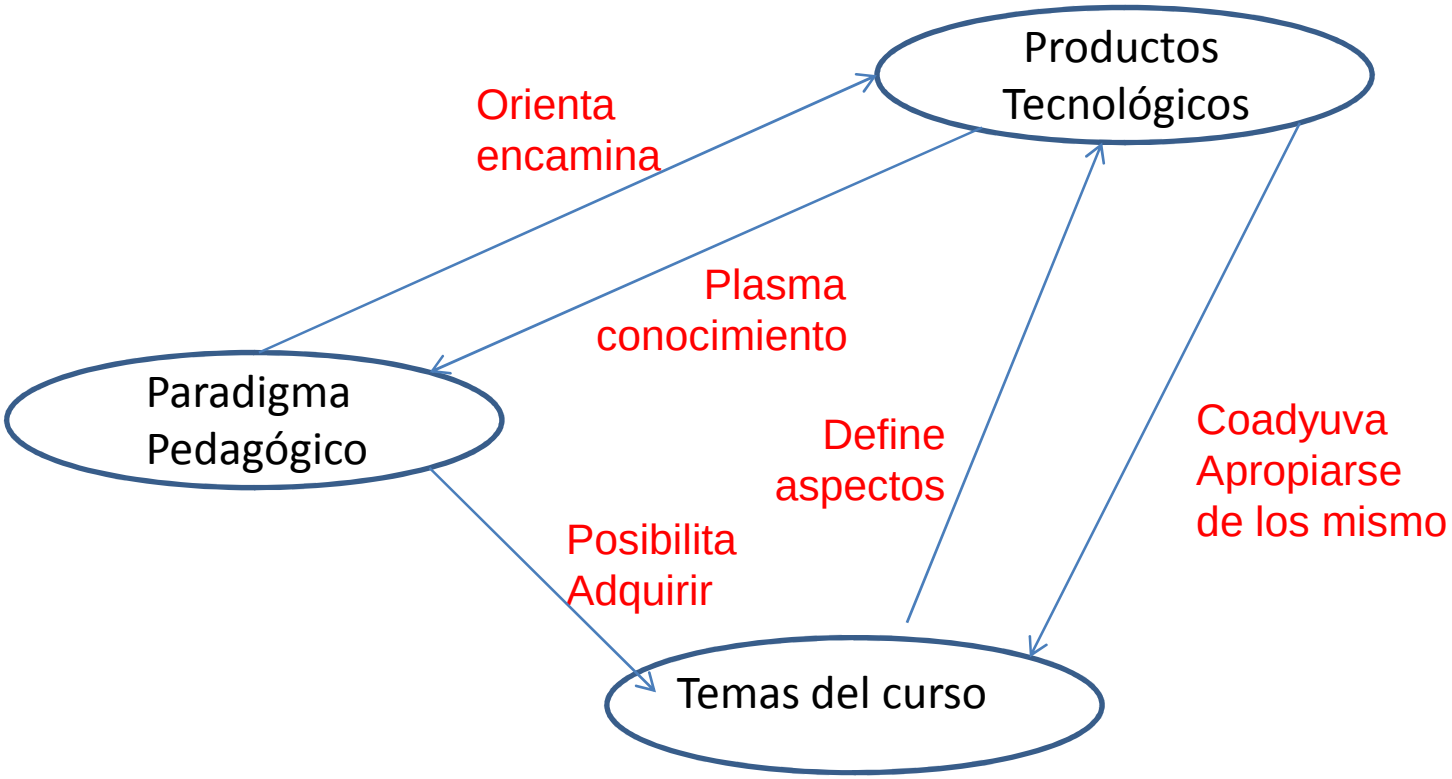
N. Nilsson, "Artificial Intelligence: a new synthesis", Morgan Haufmann Publishers, 1998.

G. Weiss, "Multi- agent System: a modern approach to distributed artificial intelligence", MIT Press, 1999.

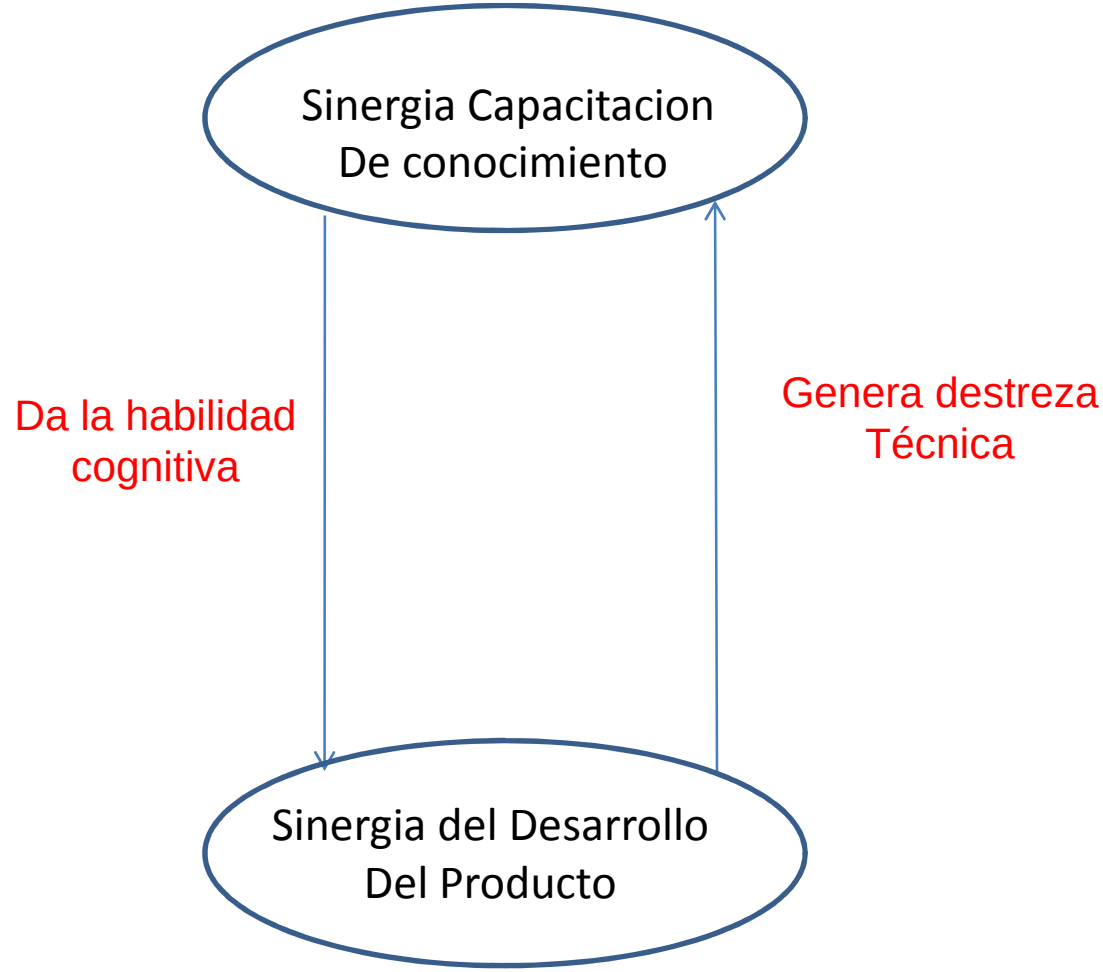
, *http://aima.cs.berkeley.edu/.* con casi 900 enlaces a páginas relacionadas con el campo de la Inteligencia Artificial

.

Dinámicas del Curso



Dinámicas del Curso



Sinergias

- Sinergia Capacitación de Conocimientos (SCC)
  - Conocimiento de base del curso
  - Constituido por los diferentes aspectos que configuran el ámbito de conocimiento
  - Todo el material en línea, internet
  - Espacios de discusión y debate

Sinergias

- Sinergia Desarrollo de Productos (SDP):
  - Definición y caracterización de un producto (obra) sobre el cual se ira plasmando el conocimiento adquirido
  - Construcción progresiva (desarrollo ágil)
  - Al final un producto (obra) con todo el contenido adquirido en el curso inmerso en él
  - Máximo de 4 personas
  - Se debe dar cuenta del recorrido del desarrollo del producto semanal (entregan informe de avance)
  - Auto-organizados (pero visibles para todos):
    - Definición de organización y roles a lo interno
    - Reglas sociales consensuadas

Recorrido Sinergia Desarrollo de Productos

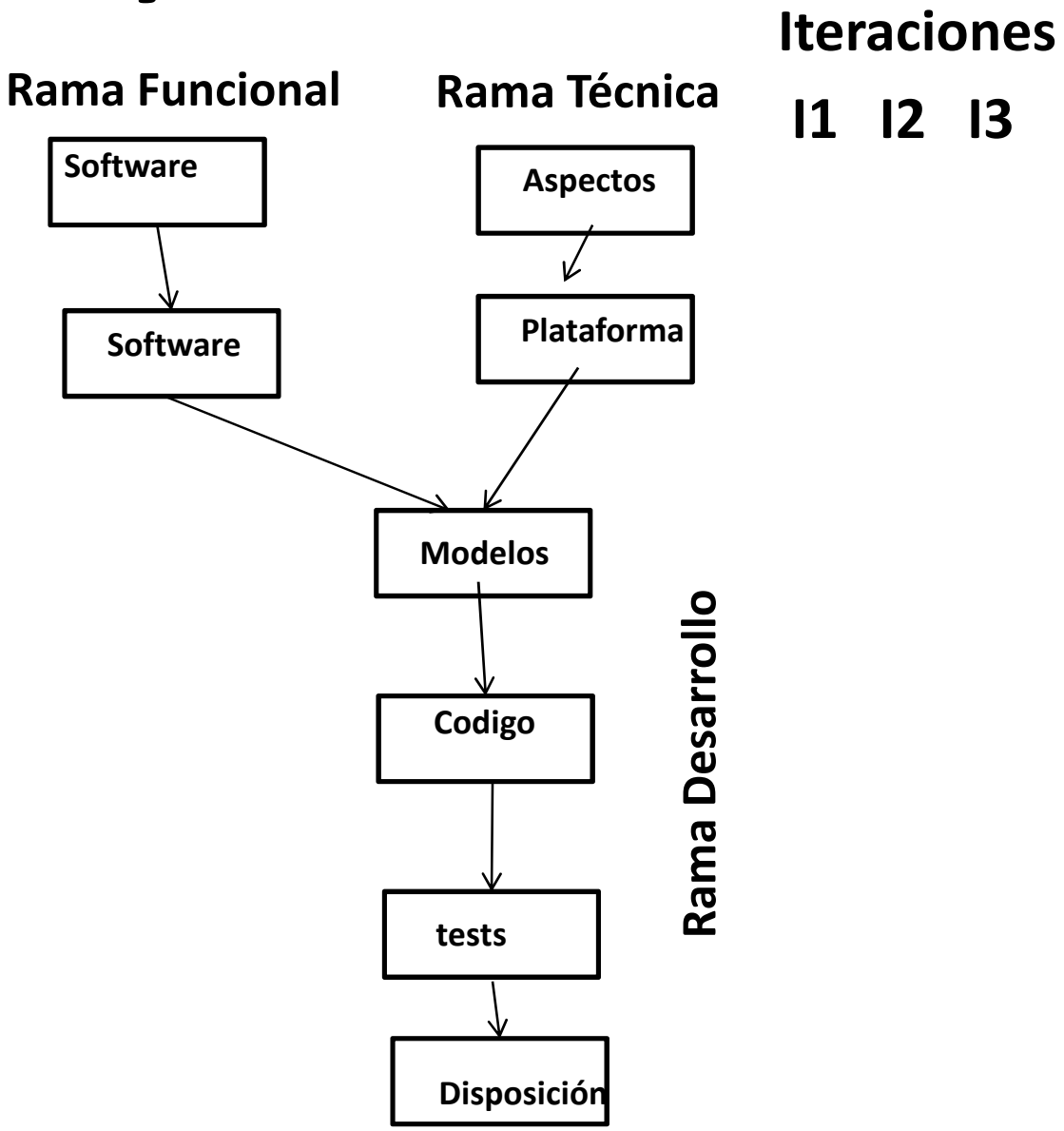
- Se seguirá metodología yPBL y MASINA
  - yPBL:
    - Ramas de diseño (funcional y técnico) y desarrollo
    - Iterativo (ágil)
    - Fases de requerimiento, análisis, diseño, implementación y tests
  - MASINA
    - Especificación de Sistemas Inteligentes
    - Modelos , diagramas de UML, y Fases
- Cada semana se avanzará en diseño y desarrollo según SCC mostrando:
  - Reutilización de componentes,
  - Herramientas de desarrollo, etc.
  - Prototipos ,informes de Avances, informes de iteraciones

yPBL

- **Metodología de aprendizaje inspirada en Ingeniería de software**
- **Permite construir aplicaciones reales de software mientras se aprende.**
- **Cada Iteración:**
  - **Cubre un tópico del curso aplicado al producto tecnológico**
  - **Se redefinen roles en los grupos, recursos usados, cronogramas**
  - **Interactuamos todos para alcanzar los objetivos de aprendizaje**

yPBL

- Requerimientos
- Análisis
- Diseño
- Implementación
- Pruebas
- Liberación



UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

Metodología MASINA

Metodología que permite especificar Sistemas Distribuidos Inteligentes.

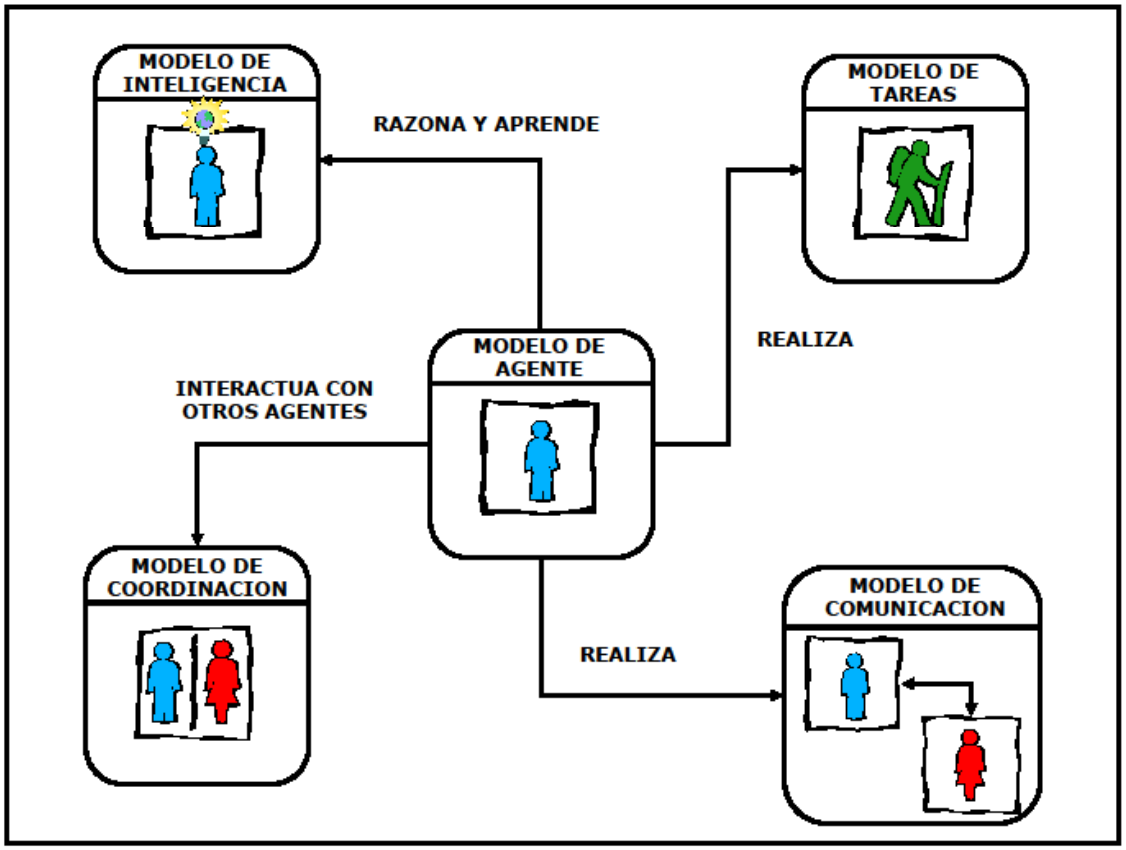
Fases

- *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 el sistema, sus tareas, sus formas de comunicación.
  - Diseño técnico del sistema.
- *Codificación y prueba*
- *Integración*
- *Operación y mantenimiento*

UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

MASINA

Modelos de MASINA



yPBL y MASINA

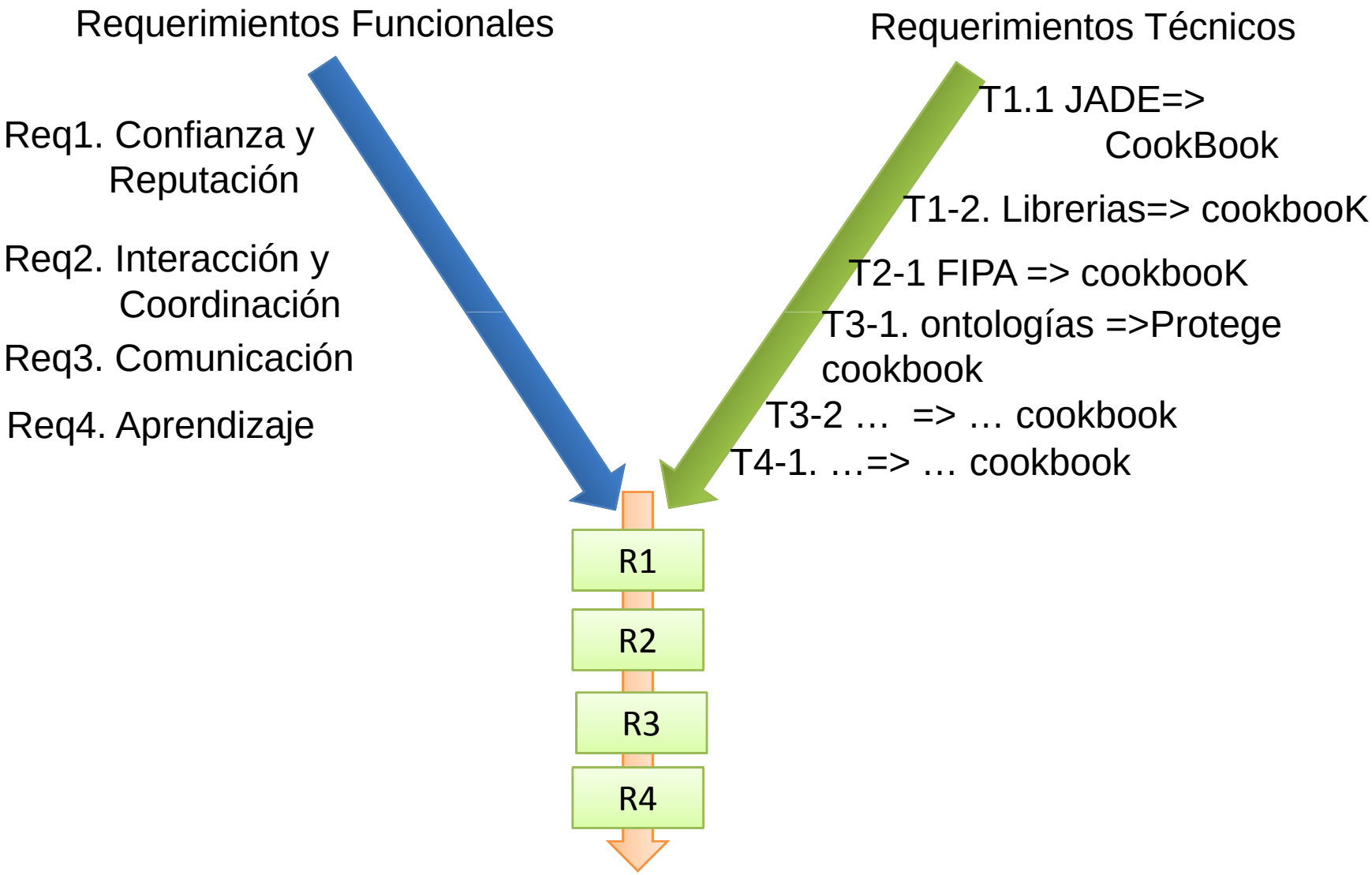
- **Metodología de base: yPBL**
- **Fase análisis de la Rama Funcional de yPBL se basa en las fases de conceptualización y de análisis de MASINA**
- **Fase de diseño de yPBL se basa en de MASINA**

Evaluación

- Conocimiento Adquirido (40%)
  - Aplicaciones, investigaciones, discusiones
  - Exámenes cortos

- Producto (60%)
  - Informes técnicos de Avances
  - Informes semanales (que se hizo en la semana, quien lo hizo, que se debió hacer, que se va a hacer)
  - Componente Tecnológico desarrollado

Desarrollo del curso



Sem	Iter	SDP	SCC	Producto
1	1	Definición del Producto	Introducción a la asignatura, tema 1.3	
2	1	Especificación de los Agentes	Temas 1.1, 1.2 y 2.1	
3	1	Modelo base SMA	Temas 2.2	
4	1	Modelo base SMA	Tema 2.2 y 2.3	
5		Revisión de Conocimiento		
6	2	Interacción entre los Agentes	Temas 3.1	Informe Técnico
7	2	Interacción entre los Agentes	Tema 3.2	
8	2	Interacción entre los Agentes	Tema 3.3	
9	3	Comunicaciones entre los Agentes	Tema 3.4	Informe Técnico.
10		Revisión de Conocimiento		
11	4	Aprendizaje en SMA	Tema 4.1	Informe Técnico
12	4	Aito-organización en SMA	Tema 4.2	
13	4	Aito-organización en SMA	Tema 4.2	
14		Modelos de rerefencia en SMA	Tema 4.3	Informe Técnico
15		Revisión de Conocimiento		.
16		Presentación Producto		

Desarrollo del Producto

- **Esencia del producto vs. objetivo del curso**
- **Esencial al curso para apropiarse del conocimiento**
- **Contenido del curso esencial para realizar el producto**
- **Cuatro plantillas:**

Plantillas	Uso
Definición del producto	Primera Semana
Informes de Avance	Semanalmente
Informes Técnicos	Final de cada iteración
CookBook	Final de cada iteración

Plantillas

Definición del Producto

- **Nombre**
- **Objetivo**
- **Descripción**
- **Alcance**
- **Conocimiento Requerido**
- **Materiales requeridos**
- **Cliente/Doliente**

Plantillas

- **Informes de Avance**
  - Planificación de la semana siguiente
  - Qué se logró en la semana
  - Quién hizo qué, dificultades y necesidades

- **Informes Técnicos**
  - Objetivo de la iteración
  - Caracterización del mismo en el producto
  - Planificación de la iteración
  - Diseño del mismo en el producto
  - Prototipo y pruebas del mismo

Plantillas

CookBook

- Resumen (Abstract)
- Palabras Claves (Keywords)
- Contribuyentes (Contributors)
- Versiones (Releases)
- Introducción (Introduction)
- Ingredientes: Definiciones y Terminología (The ingredients: Definitions and terminology)
  - Ingrediente 1 (Ingredient 1)
- Recetas (Recipes)
  - Receta 1: Una primera receta (Recipe1: A first recipe (e.g. a HelloWorld recipe))
    - Paso 1: descripción paso 1 (Step1: short description of step 1)
- Documentación Recomendada (Recommended documentation)
- Referencia 1 (Reference 1)
- Retroalimentación (Feedback)

Agentes

Jose Aguilar
Cemisid, Facultad de Ingeniería
Universidad de los Andes
Mérida, Venezuela
aguilar@ula.ve

Agente

Es un sistema computacional que está *situado en un entorno*, que es capaz de realizar *acciones autónomas* flexibles en ese entorno para alcanzar sus objetivos

- Aspectos:
  - SU ESTRUCTURA (ARQUITECTURA)
  - SUS ACCIONES (COMPORTAMIENTO)

Aquitectura+programa

UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

Agentes

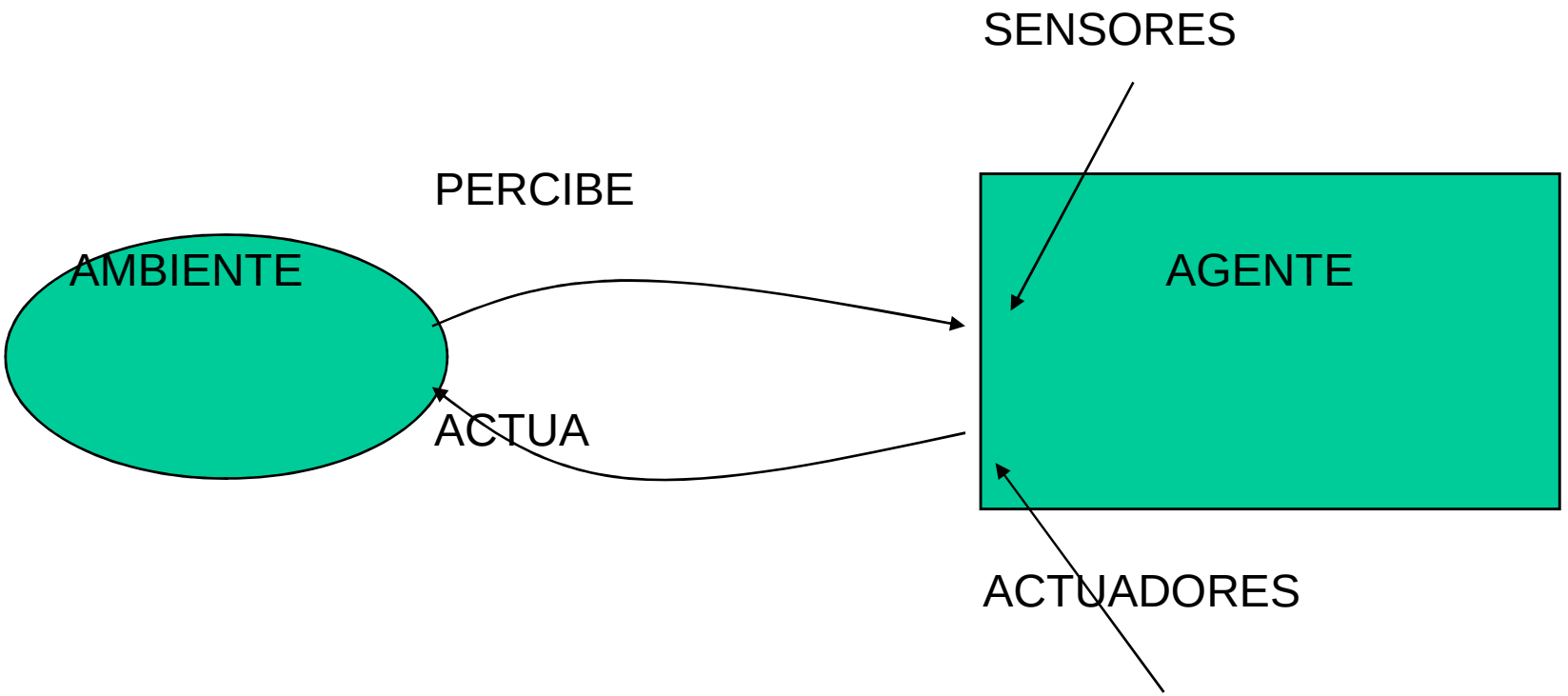
Los agentes de software son capaces de *decidir por sí mismos* qué hacer en una determinada situación. *Mantienen información* acerca de su entorno, y *toman decisiones* en función de su percepción del estado de dicho entorno, sus experiencias anteriores, y los objetivos que tienen planteados. Además, los agentes *pueden comunicarse* con otros agentes para colaborar y alcanzar objetivos comunes. [Denney 2008]

Agente

Es cualquier cosa que pueda percibir su ambiente a través de *sensores*, y actuar sobre él a través de *actuadores*

- Agente Humano: ojos, oídos, manos, brazos, etc.,
- Agente Robot: cámaras, sensores y varios motores actuadores

Agentes



Agente

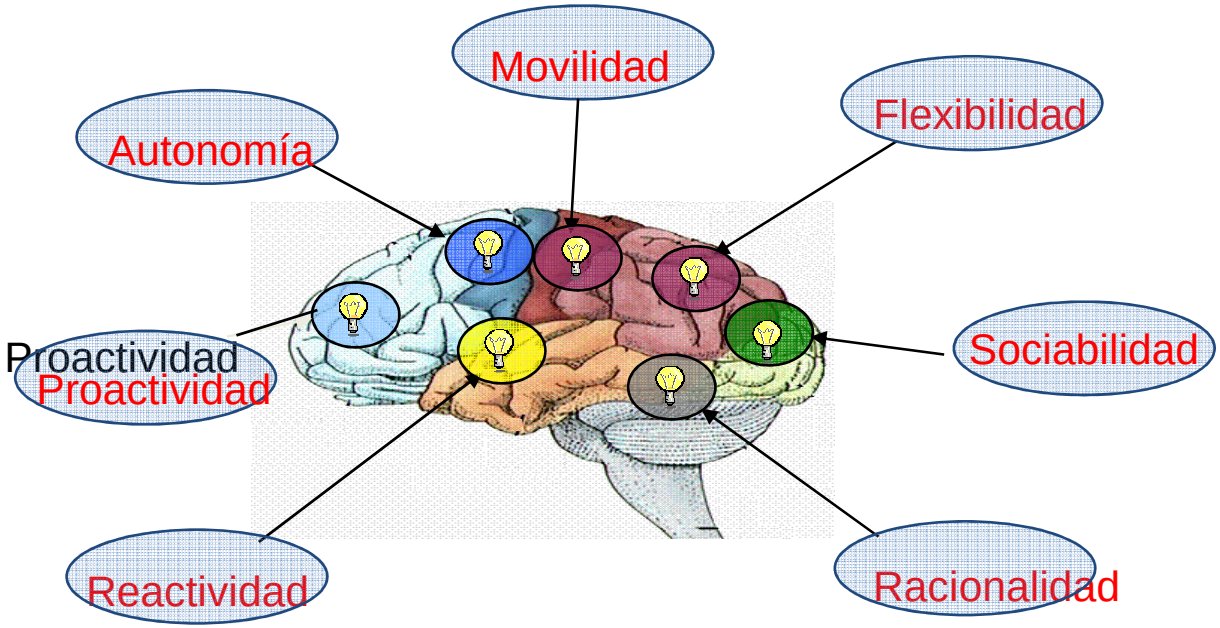
- UN *AGENTE* ES TODA ENTIDAD FISICA O VIRTUAL QUE:
  1. ES CAPAZ DE INTERACTUAR CON SU AMBIENTE
  2. PUEDE COMUNICARSE DIRECTAMENTE CON OTROS AGENTES
  3. TIENE UN CONJUNTO DE DESEOS (P.E.: BAJO LA FORMA DE OBJETIVOS INDIVIDUALES O FUNCIONES DE SATISFACCION, QUE BUSCA OPTIMIZAR)
  4. POSEE RECURSOS PROPIOS
  5. ES CAPAZ DE PERCIBIR SU AMBIENTE
  6. TIENE UNA REPRESENTACION PARCIAL DE SU AMBIENTE
  7. POSEE CIERTAS COMPETENCIAS Y DA SERVICIOS
  8. PUEDE REPRODUCIRSE
  9. SU COMPORTAMIENTO TRATA DE SATISFACER SUS OBJETIVOS

Agentes

- Ejemplos de Agentes:
  - Agente Humano: ojos, oídos, manos, brazos, etc.,
  - Agente Robot: cámaras, sensores y varios motores actuadores

propiedades

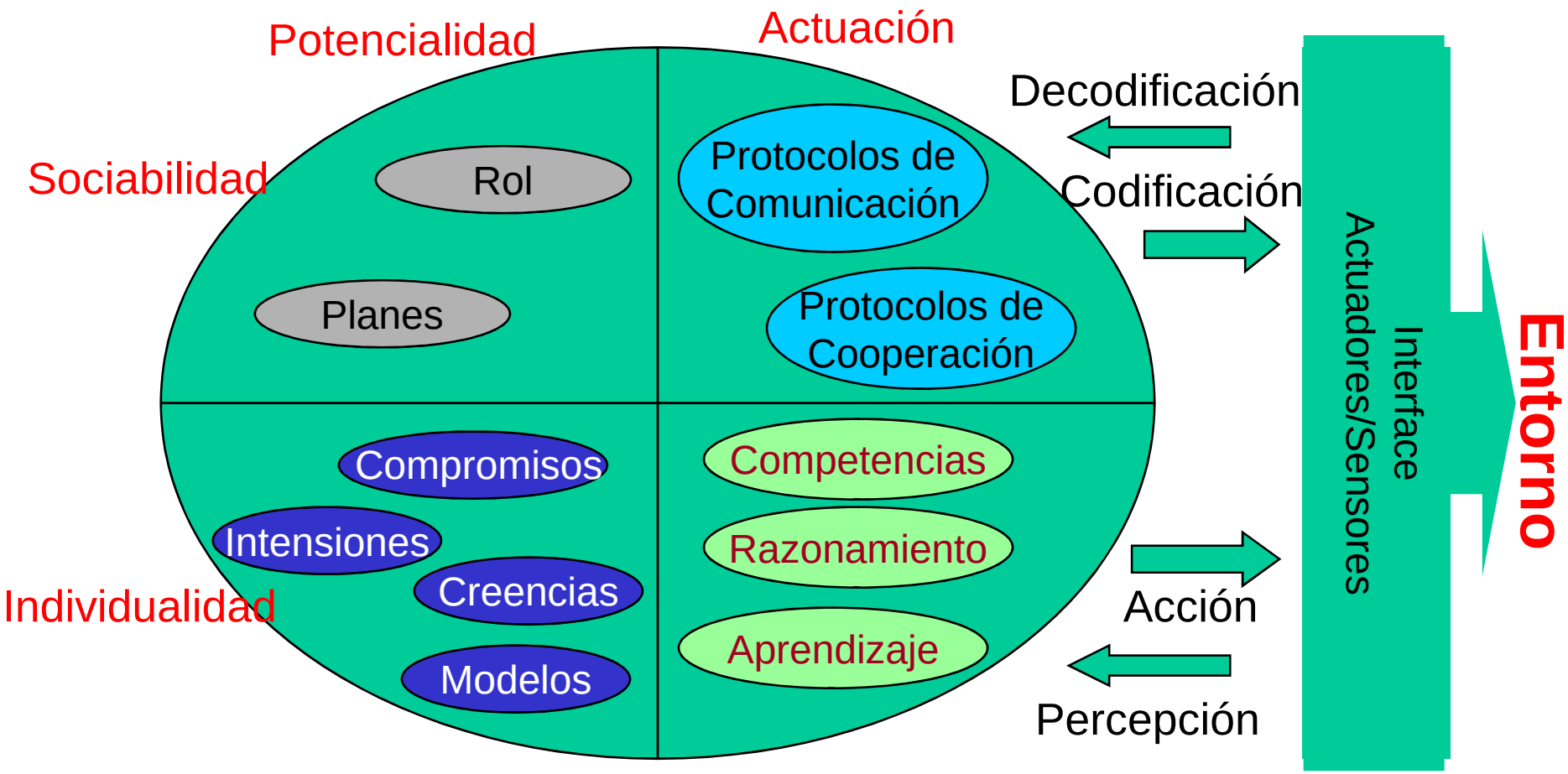
de un agente



Propiedades de los Agentes

- Adaptabilidad
- Movilidad
- Persistencia
- Autonomía
- Reactividad
- Proactividad
- Sociabilidad (Comunicativo)
- Razonabilidad
- Racionalidad
- Flexibilidad

Arquitectura de los Agentes



Esqueleto de un Agente

Procedimiento de base:

1. Percibo (Actualiza *Memoria*)
2. Decido (Escoge *Acción*)
3. Actúo (Actualiza *Memoria*)

Descripción práctica de un agente:

Sus Tareas.
Sus Conocimientos.
Su Comunicación

UNIVERSIDAD DE LOS ANDES
MERIDA VENEZUELA

Otros aspectos importantes a considerar en los Agentes

- Mecanismos para resolver un problema
- Mecanismo para planificar sus actividades
- Mecanismos para representar el conocimiento
- Mecanismo de razonamiento
- Mecanismos de aprendizaje
- Mecanismos de percepción

Tipos de Agentes

- BASADA EN CAPACIDADES:
  - PURAMENTE COMUNICANTE: 1, 2, 3, 4, 6, 7, 9
  - PURAMENTE SITUADO: 1, 3, 4, 5, 6, 7, 8, 9
  - …
- BASADA EN *MODOS DE CONDUCTA*
  - DIRIGIDA POR OBJETIVOS PRECISOS (OP)
  - DIRIGIDO POR LAS PERCEPCIONES (P)

Tipos de Agentes (según toma de decisiones)

- **Reactivos:** función situación-acción, Mecanismos de reacción. Sin planificación. Agente que actúa por reflejos. Entorno subsimbólico (capacidades sensoriales y motrices).
  - Acción: S->A
- **Con Percepción:**
  - ver: S->P
  - Acción: P->A
- **Con Estados:**
  - ver: S->P
  - Acción: E->A
  - próximo: ExP->E

S: Edos. del ambiente

A: posibles acciones

P: Percepciones

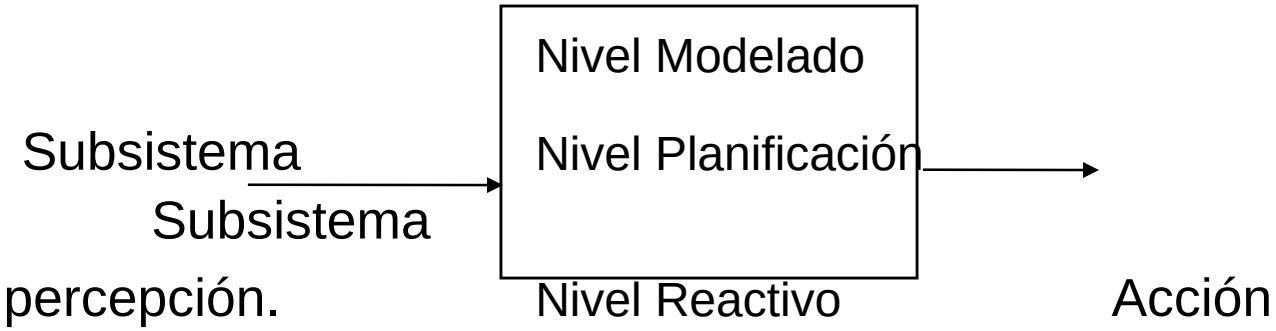
E: Estados Internos

Tipos de Agentes (según toma de decisiones)

- **Basados en deducción lógica**
  - ver: S->P
  - Acción: D->A
    lógicas
  - próximo: DxP->D

D: BD de sentencias

- **Arquitectura multinivel:** depende de la interacción entre varios elementos que razonan sobre distintas abstracciones del entorno



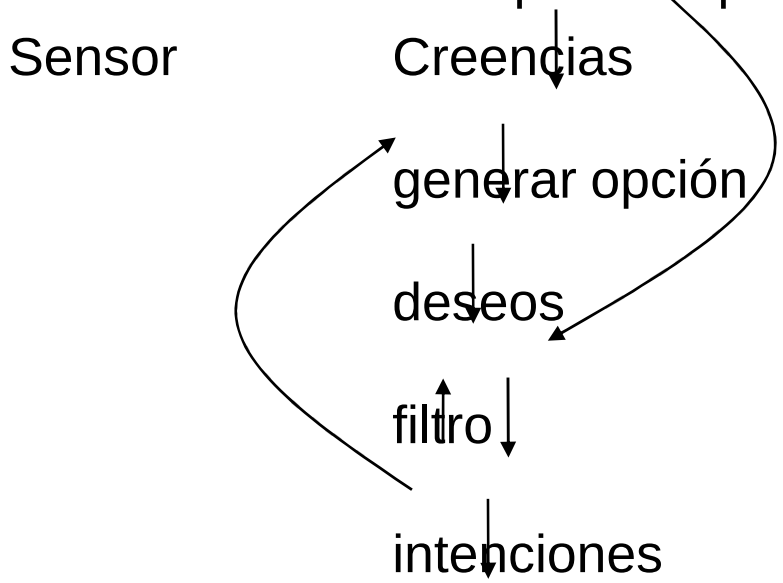
Tipos de Agentes (según toma de decisiones)

- **Agentes BDI (***Belief, Desires and Intentions***)** depende de la manipulación de estructuras de datos que representan las creencias, deseos e intenciones del agente

  - **Creencias:** Modelan el estado del Mundo.

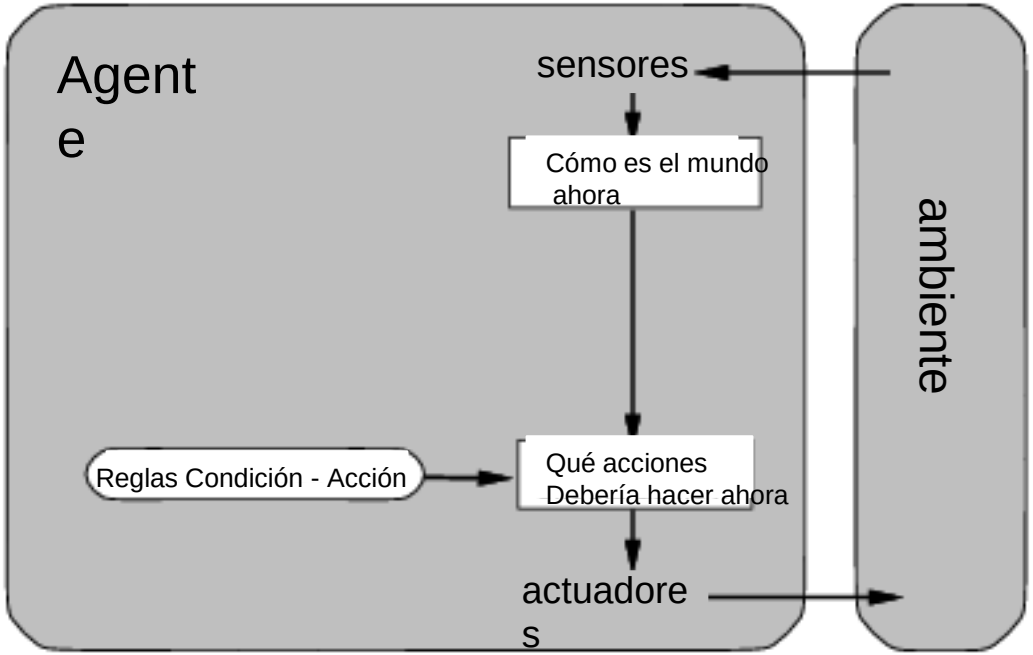
  - **Deseos**: Permiten la elección de estados posibles del mundo.

  - **Intenciones**: Son el compromiso para alcanzar un cierto estado



Agente Reflexivo

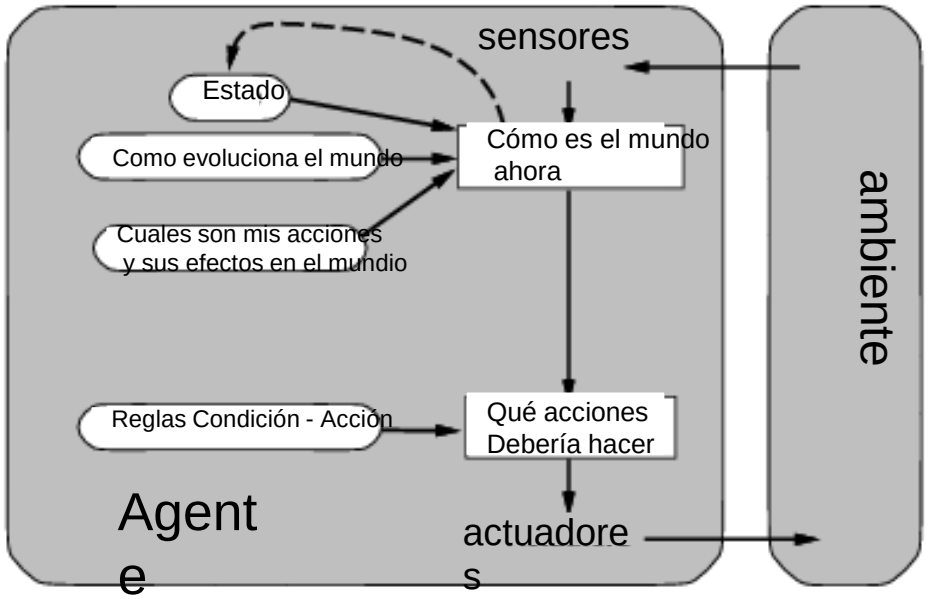
1. Interpreta (percepción)
2. Determina regla (si < -- > entonces <-->)
3. Actúa (según regla)



Agente Reflexivo con estado interno

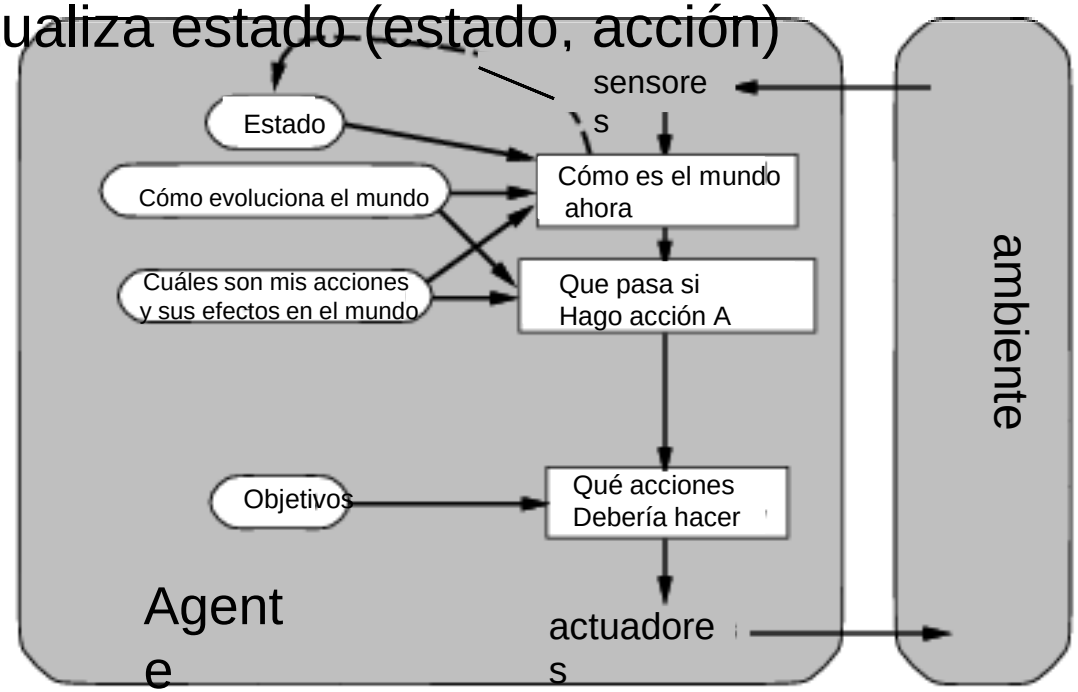
1. Actualiza estado (estado, percepción)
2. Determina regla (estado, reglas)
3. Actúa (según regla)
4. Actualiza estado (estado, acción)

Requiere de modelos del mundo!!



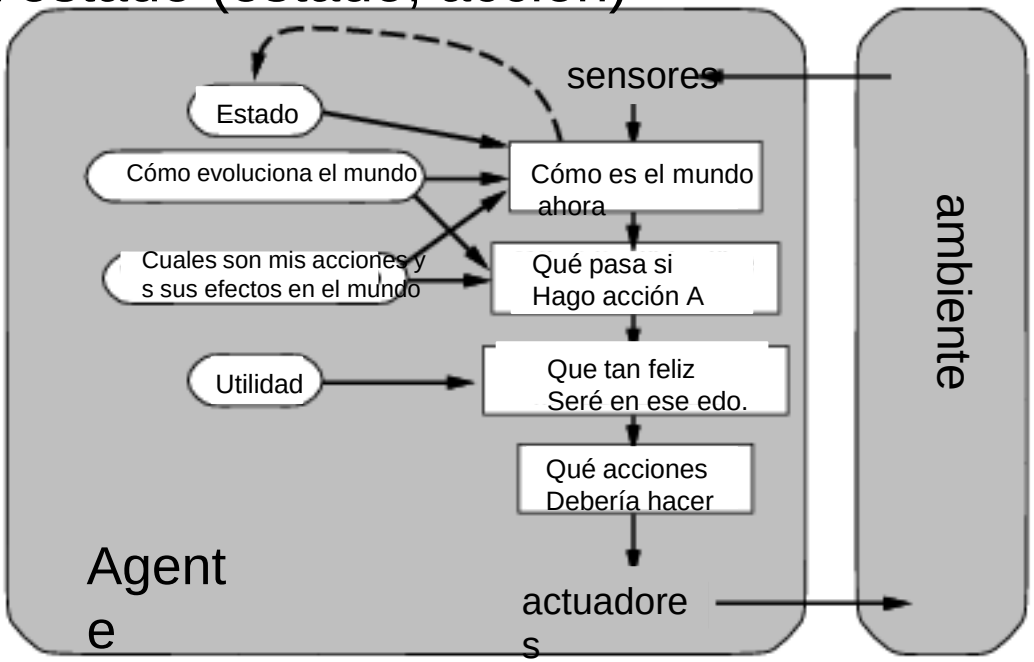
Agente con objetivos

1. Actualiza Memoria (estado, según lo que percibió)
2. Determina acción (estado, qué yo quisiera cumplir (objetivos): *búsqueda y/o planificación*)
3. Actúa (Acción inmediata, o la que corresponda según la secuencia obtenida por el proceso de búsqueda y/o plan.)
4. Actualiza estado (estado, acción)

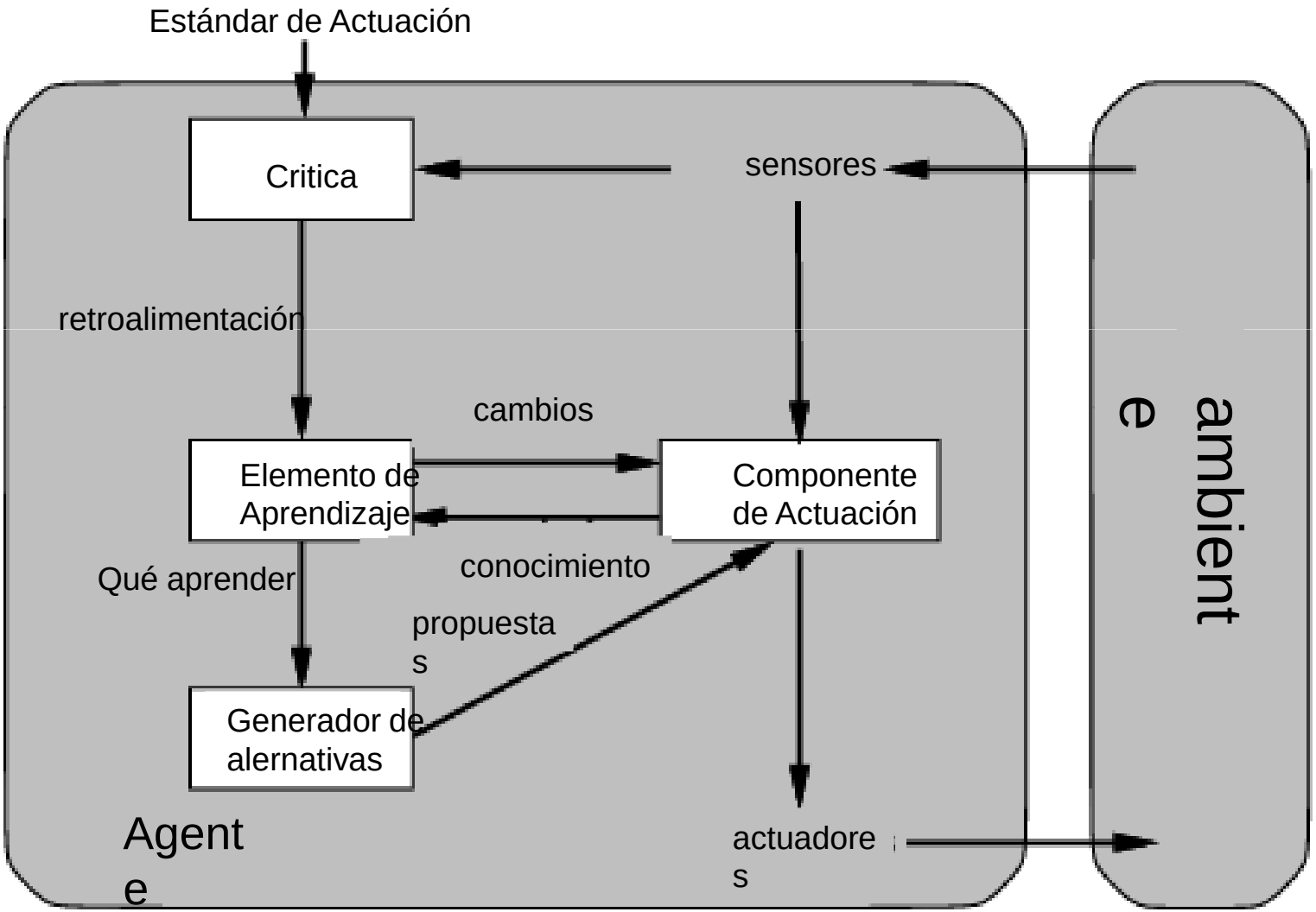


Agente con función de utilidad

1. Actualiza Memoria (estado, según lo que percibió)
2. Determina acciones (estado, qué yo quisiera cumplir (objetivos))
3. Selecciona acción (acciones, función de utilidad)
4. Actúa (según acción seleccionada)
5. Actualiza estado (estado, acción)



Agente que aprende



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

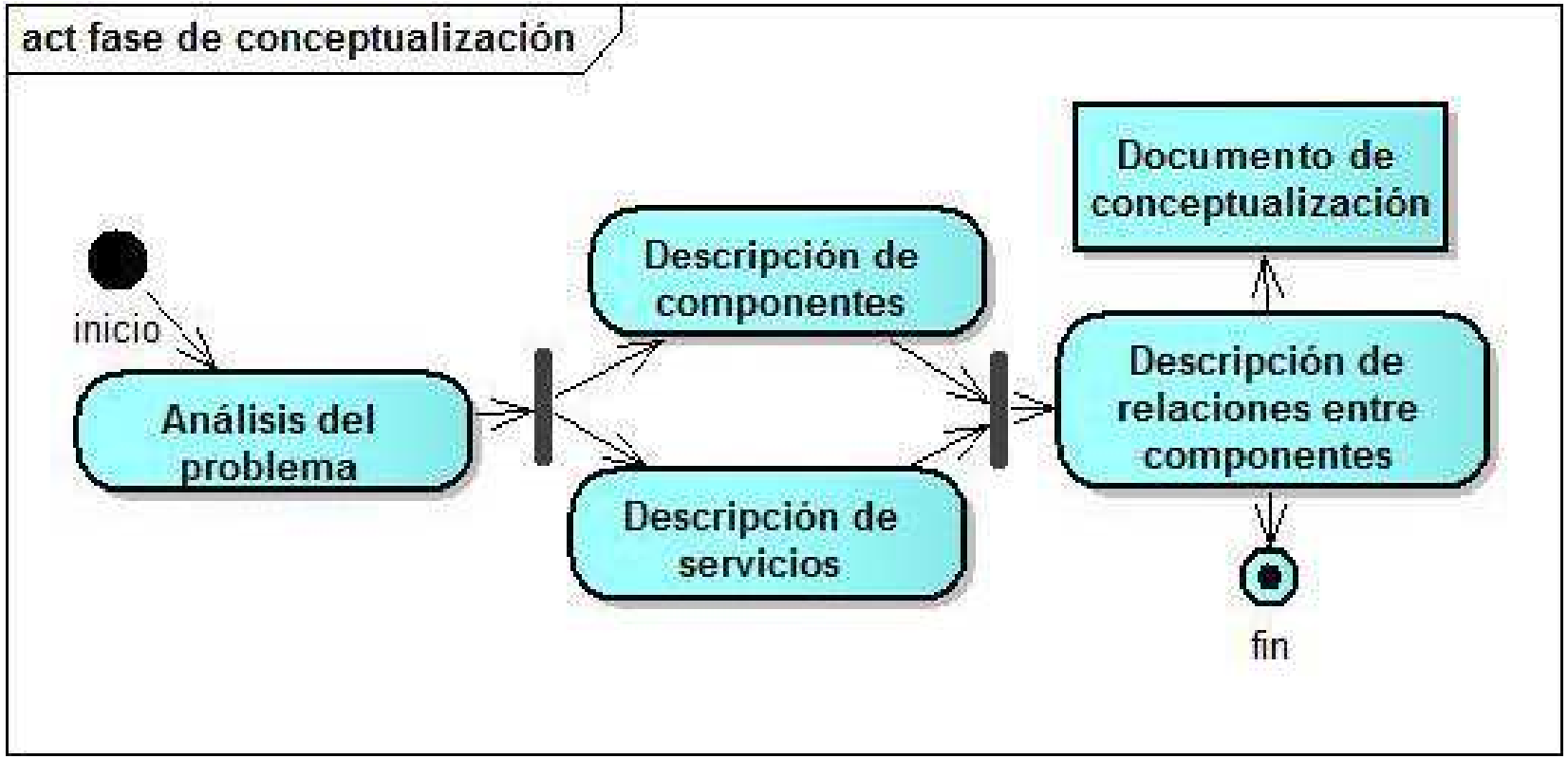
Metodología que permite especificar Sistemas Multi-agentes, la cual es una extensión de MAS-CommonKADS.

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 Conceptualización



UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

MASINA

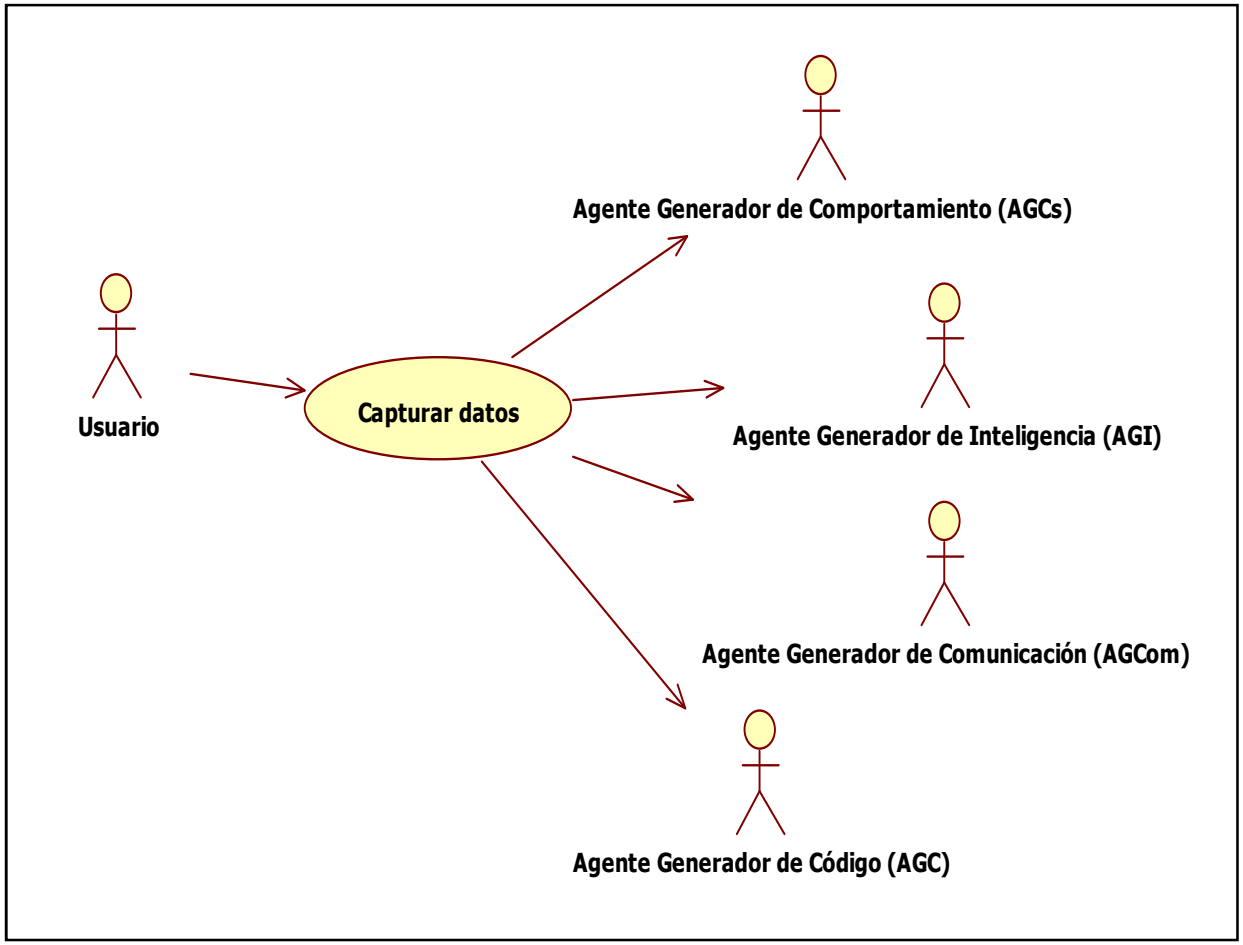
- El producto de esta fase es un documento de conceptualización:
  - La descripción de los componentes (agentes) del sistema,
  - La especificación de los servicios y de las actividades para prestar los servicios ofrecidos por cada componente del sistema
  - La descripción general de las relaciones entre los componentes del sistema.
- Para eso se usan: casos de uso y diagramas de actividades

UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

MASINA

Fase de Conceptualización

Casos de uso

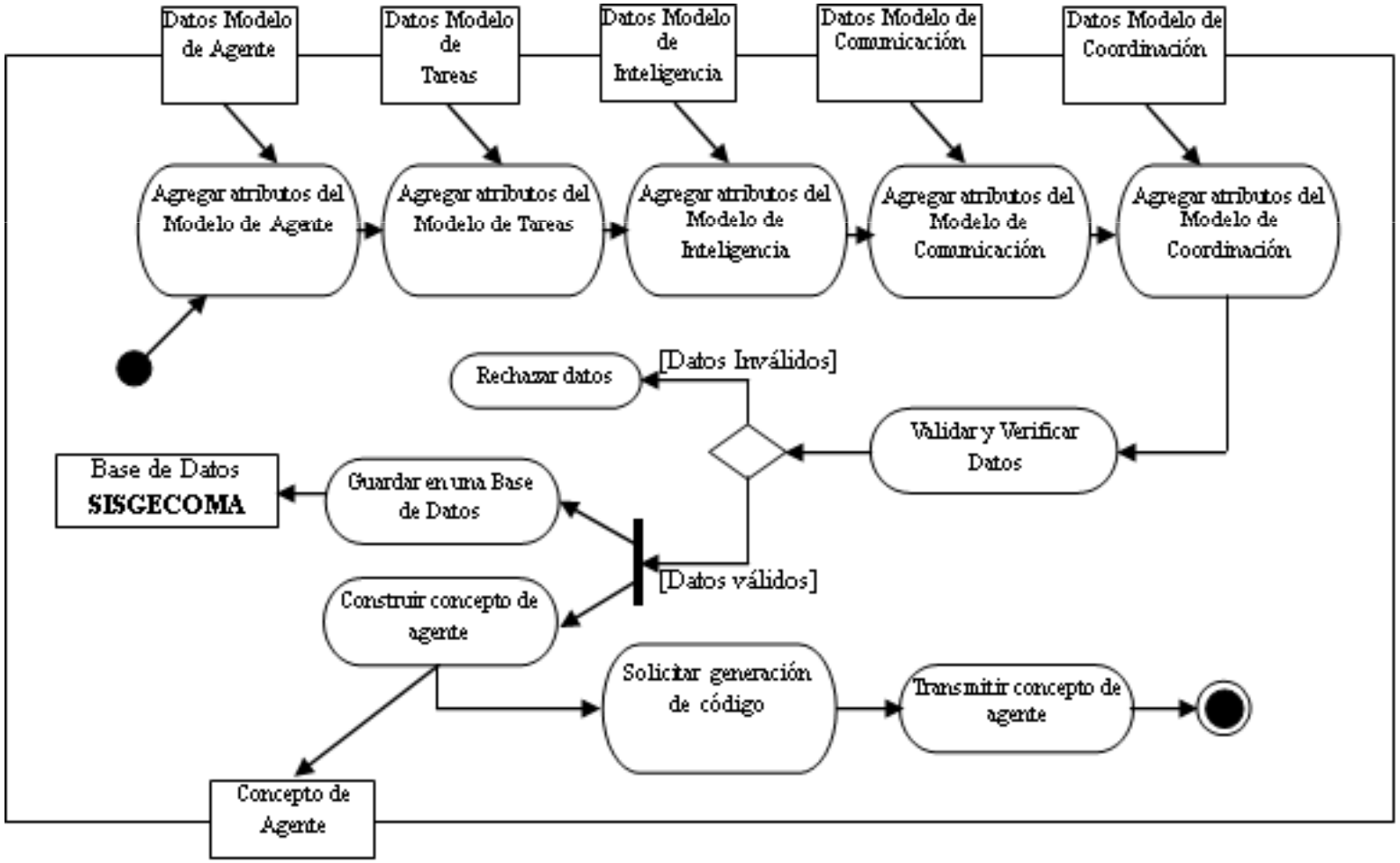


UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

MASINA

Fase de Conceptualización

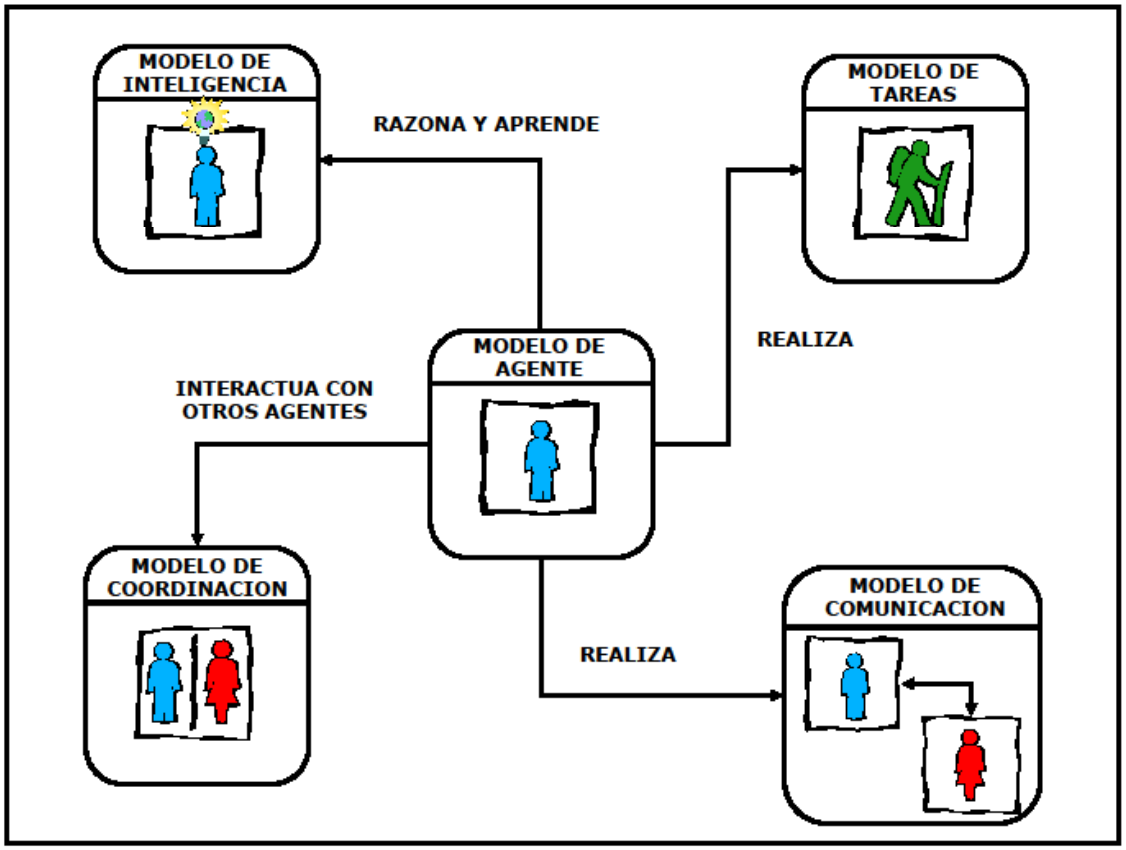
Diagrama de actividades



UNIVERSIDAD DE LOS ANDES
MERIDA VENEZUELA

MASINA

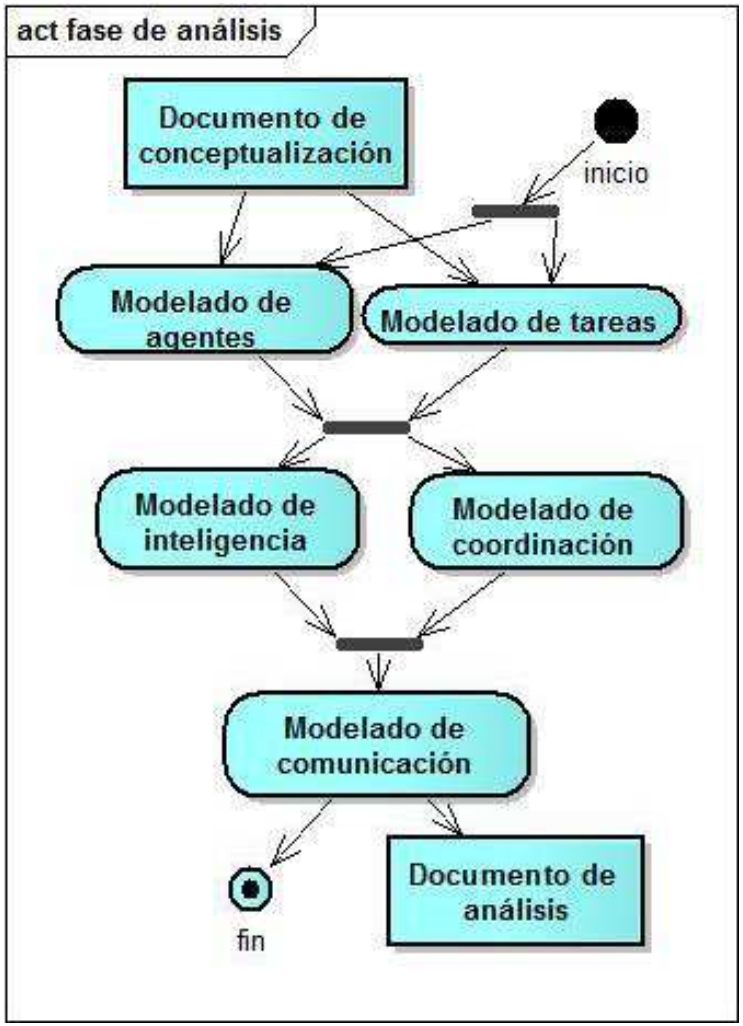
Fase de Análisis



UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

MASINA

Fase de Análisis



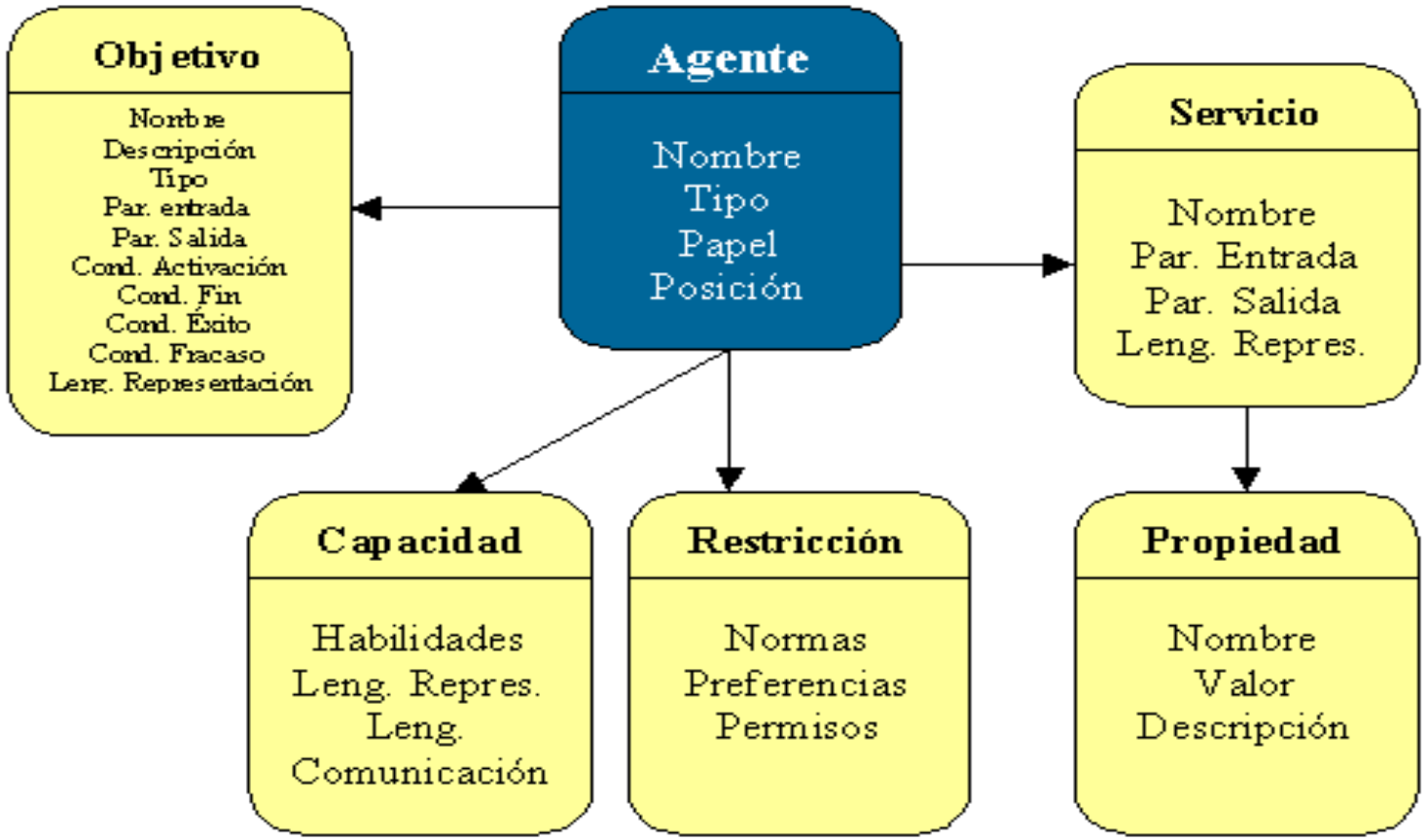
UNIVERSIDAD DE LOS ANDES
MERIDA VENEZUELA

MASINA

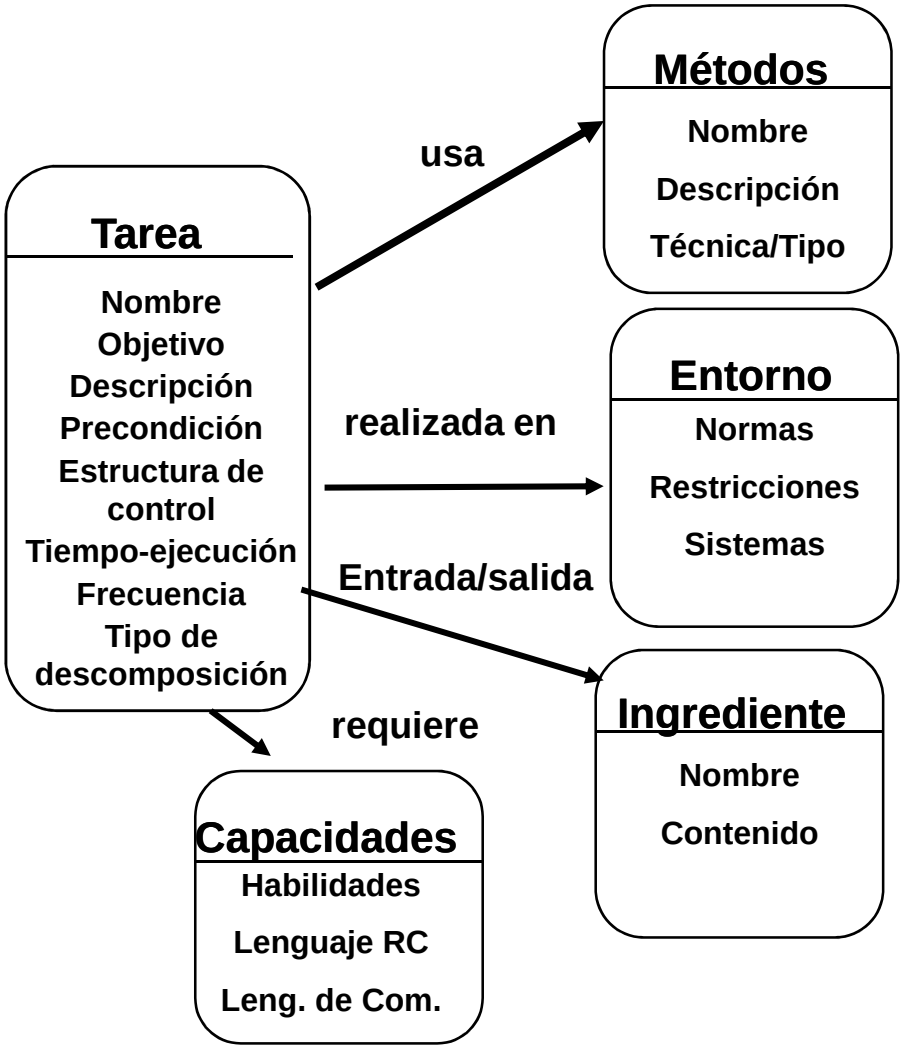
- El producto de esta fase es un documento de análisis contentivo de:
  - Modelo de agentes,
  - Modelo de tareas,
  - Tabla relación Agentes-Tareas
  - Modelo de inteligencia
  - Modelo de Coordinación/conversación
  - Diagrama de Interacción
  - Modelo de comunicación.

UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

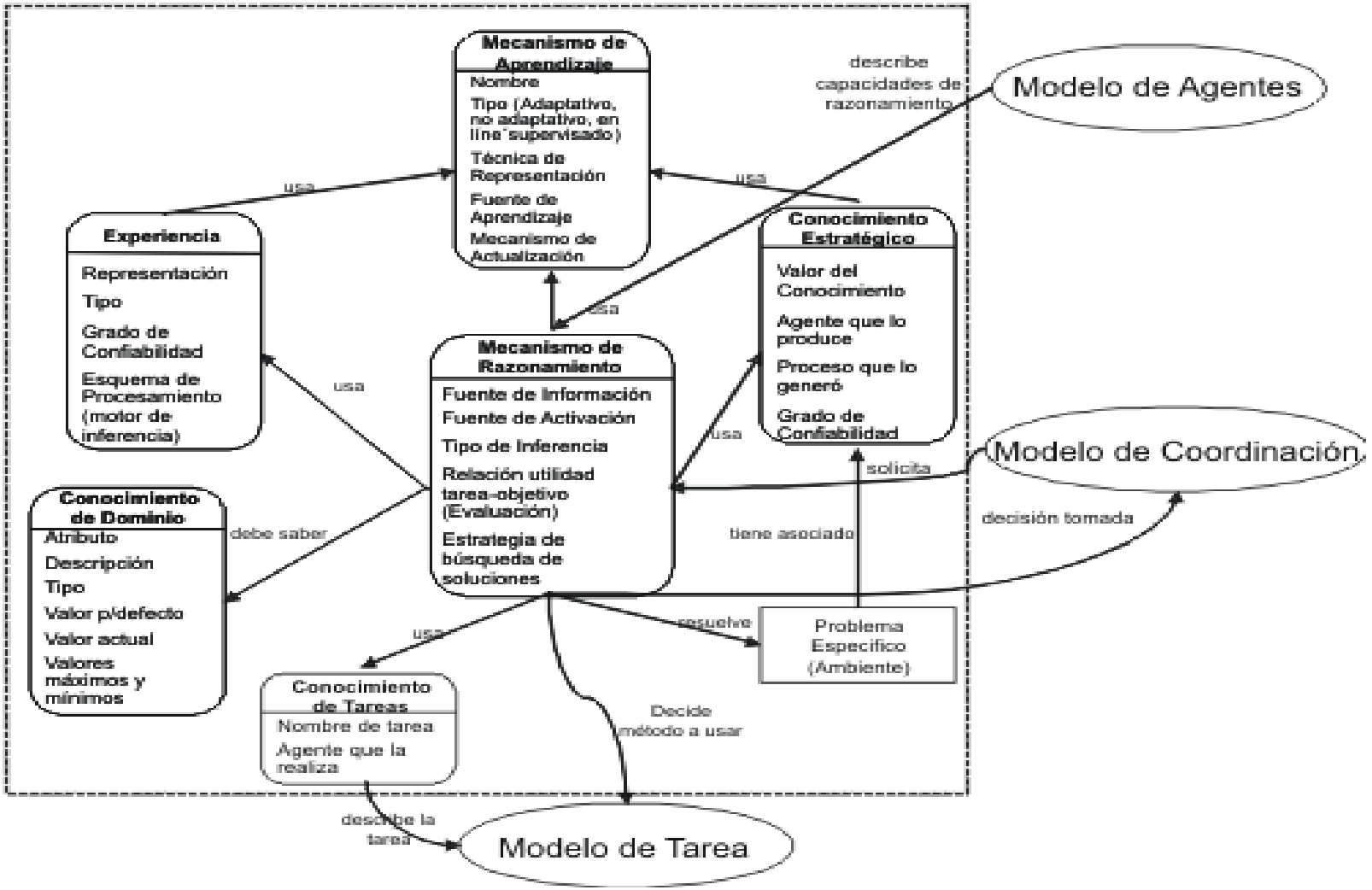
Modelo de Agente



Modelo de Tareas

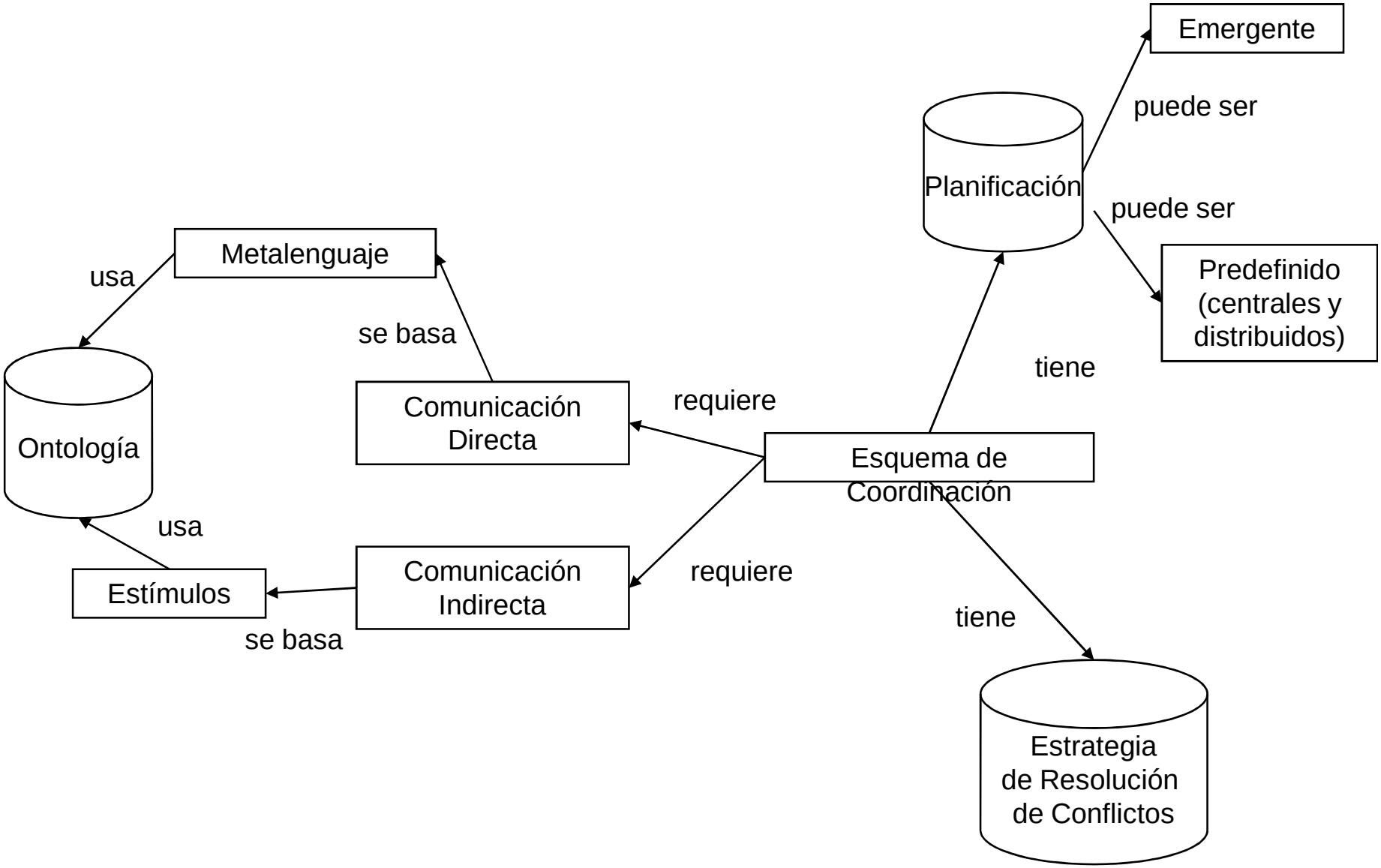


Modelo de Inteligencia

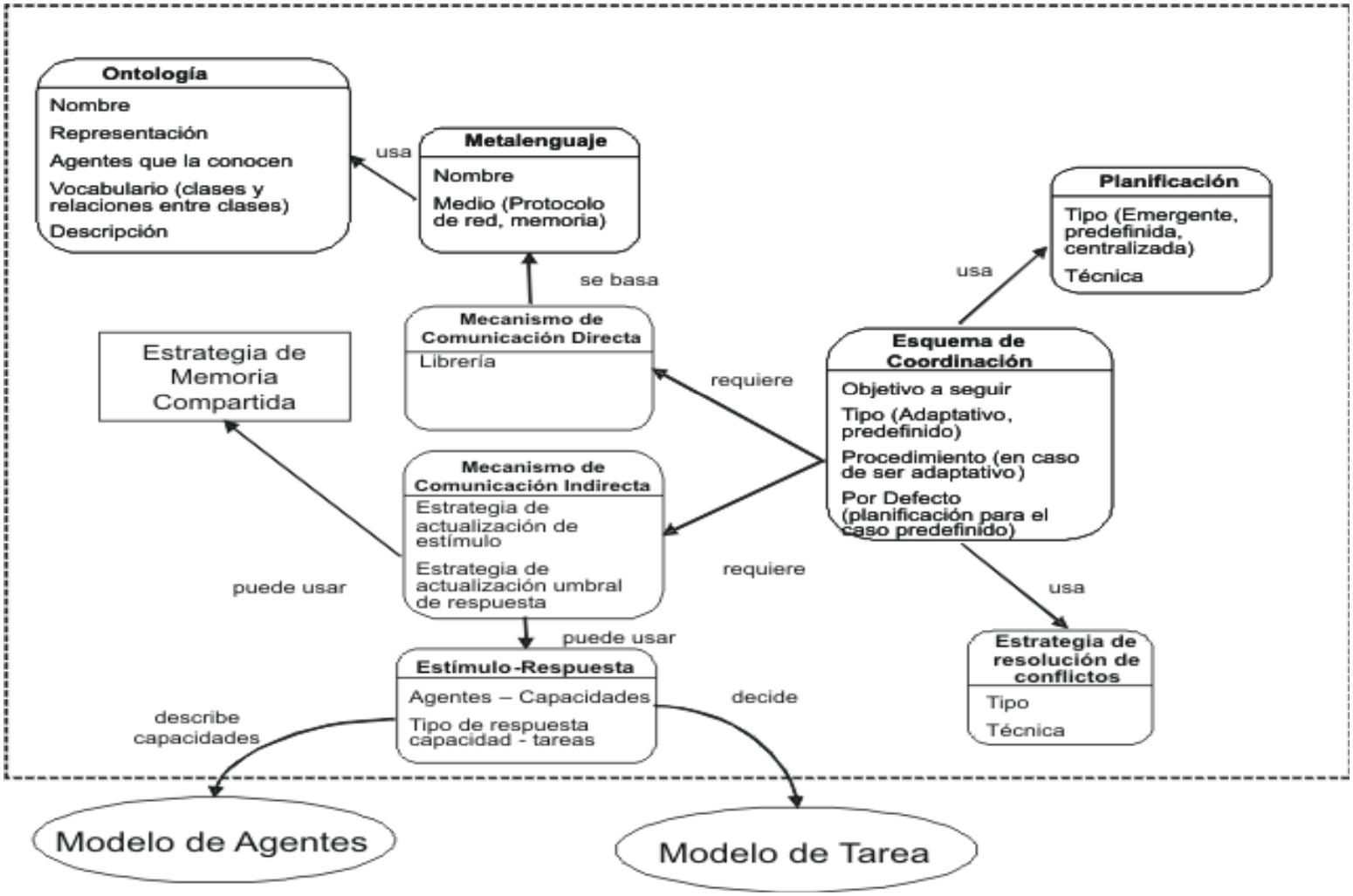


UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

Problema de Coordinación



Modelo de Coordinación



Modelo de Coordinación y Comunicación

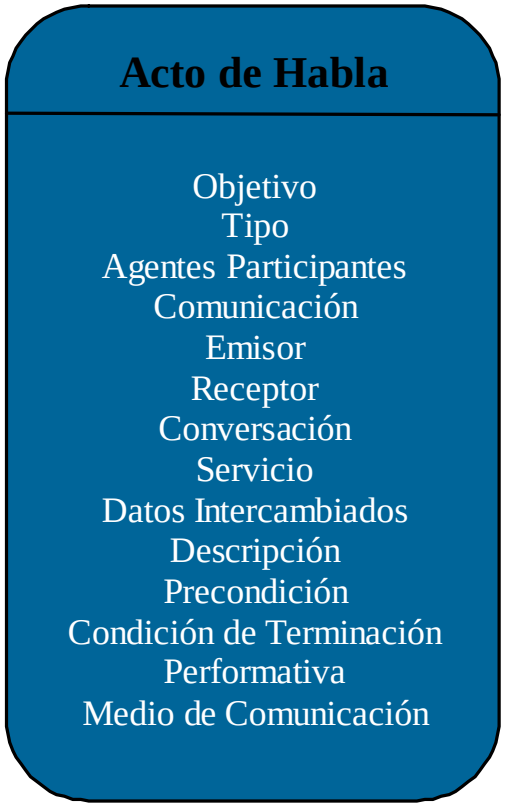
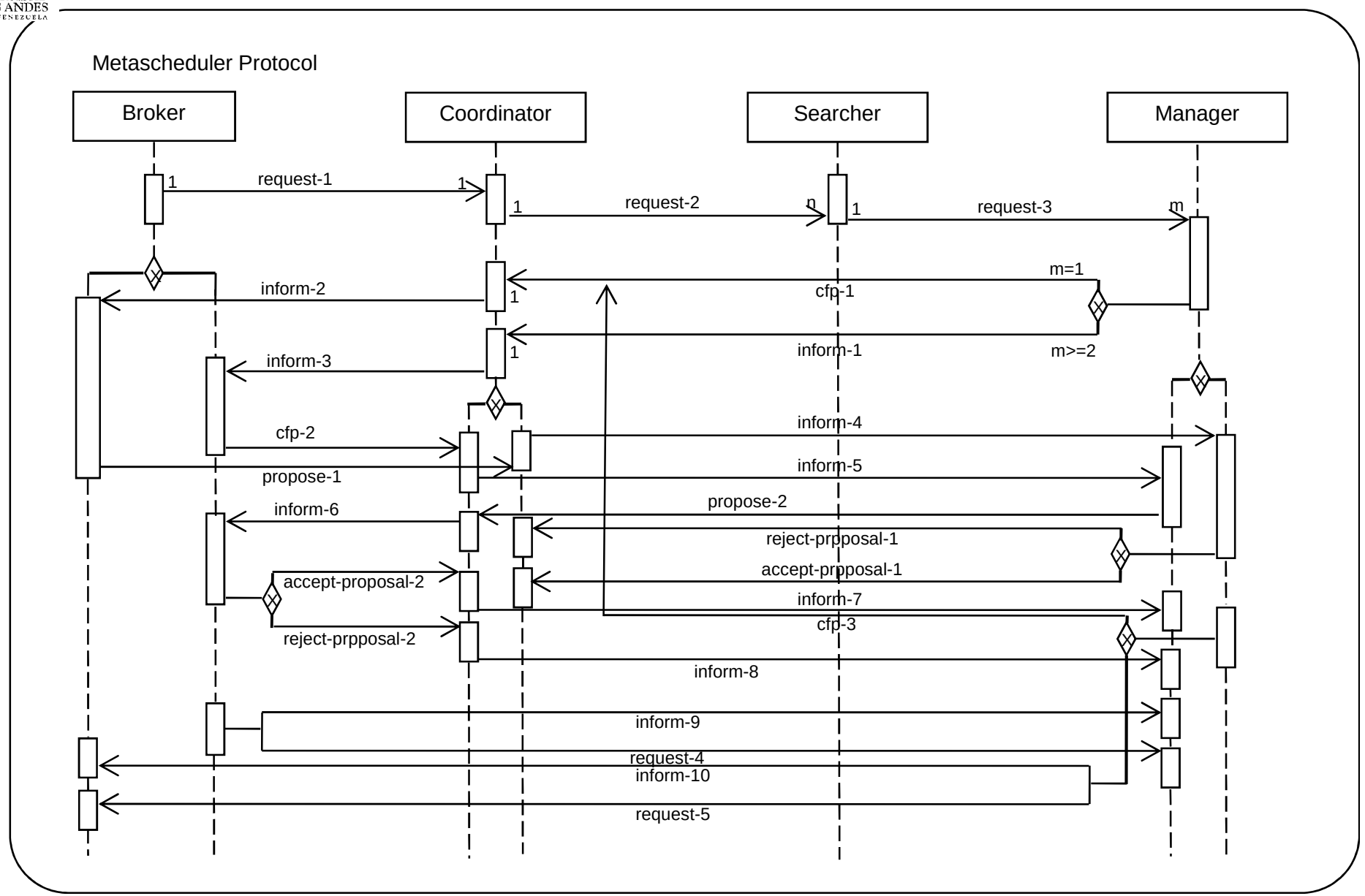


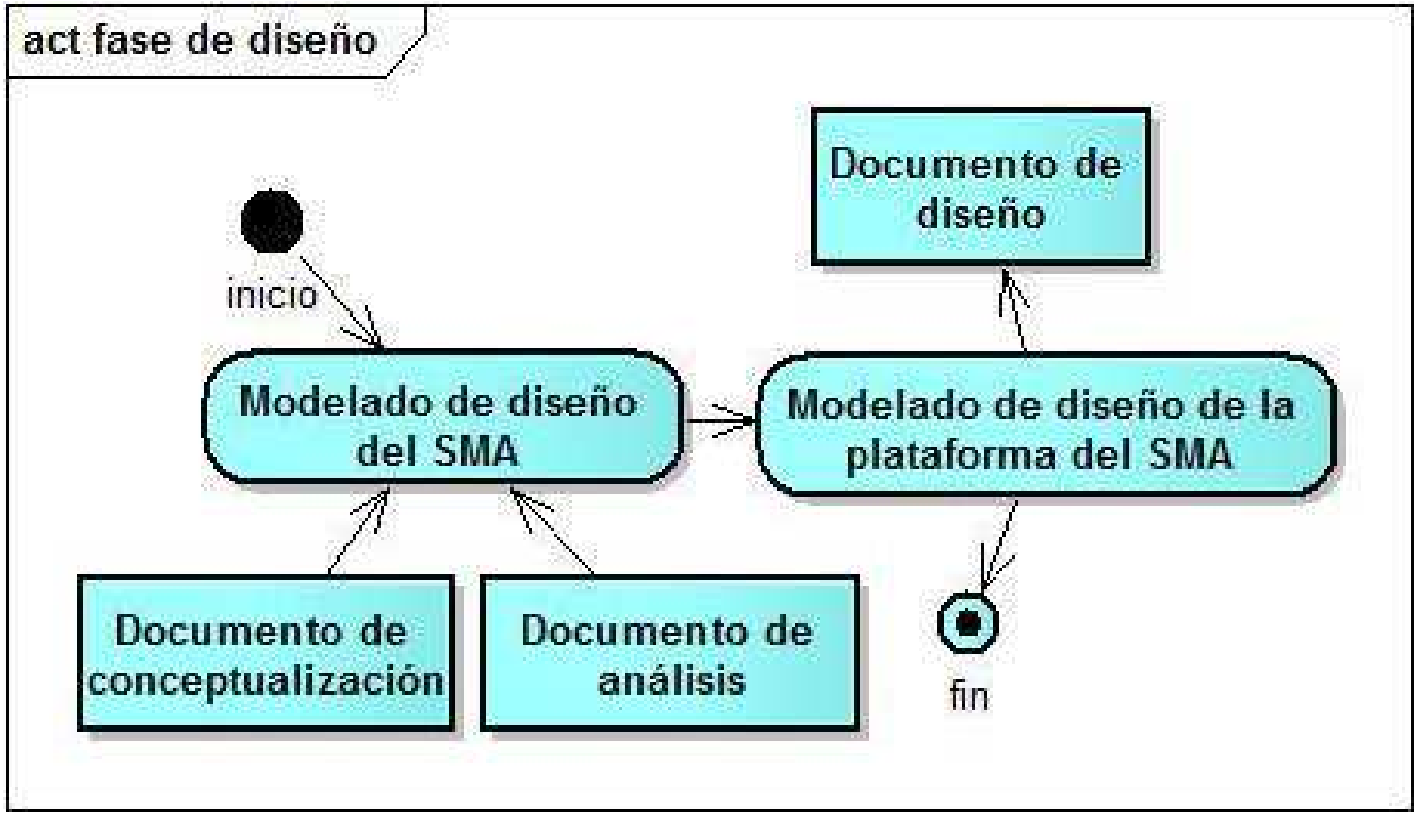
Diagrama Interacción: Protocolo de interacción FIPA



UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

MASINA

Fase de diseño



UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

MASINA

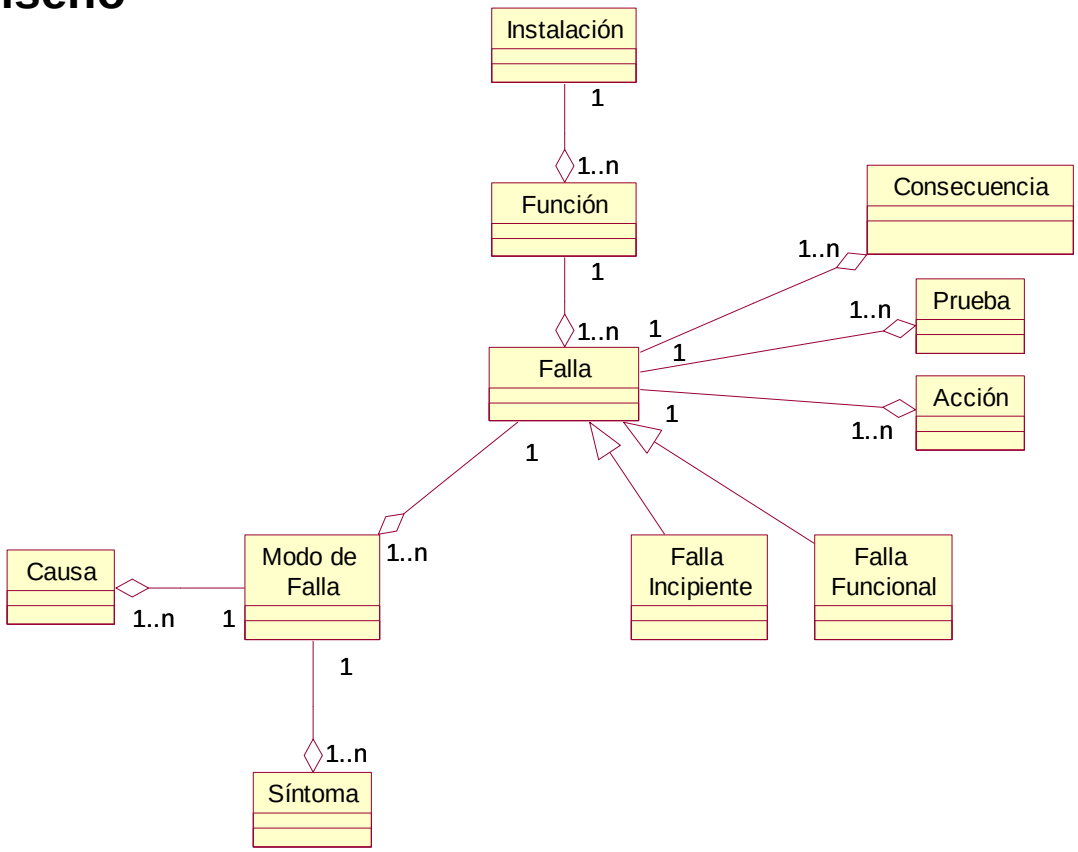
- El producto de esta fase es un documento que especifica:
  - El universo de clases del sistema.
  - Para cada una de las clases descritas en el universo de clases su especificación formal.
  - Para cada uno de los métodos que componen las clases su especificación formal.

Nosotros particularmente usamos TDSO

UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

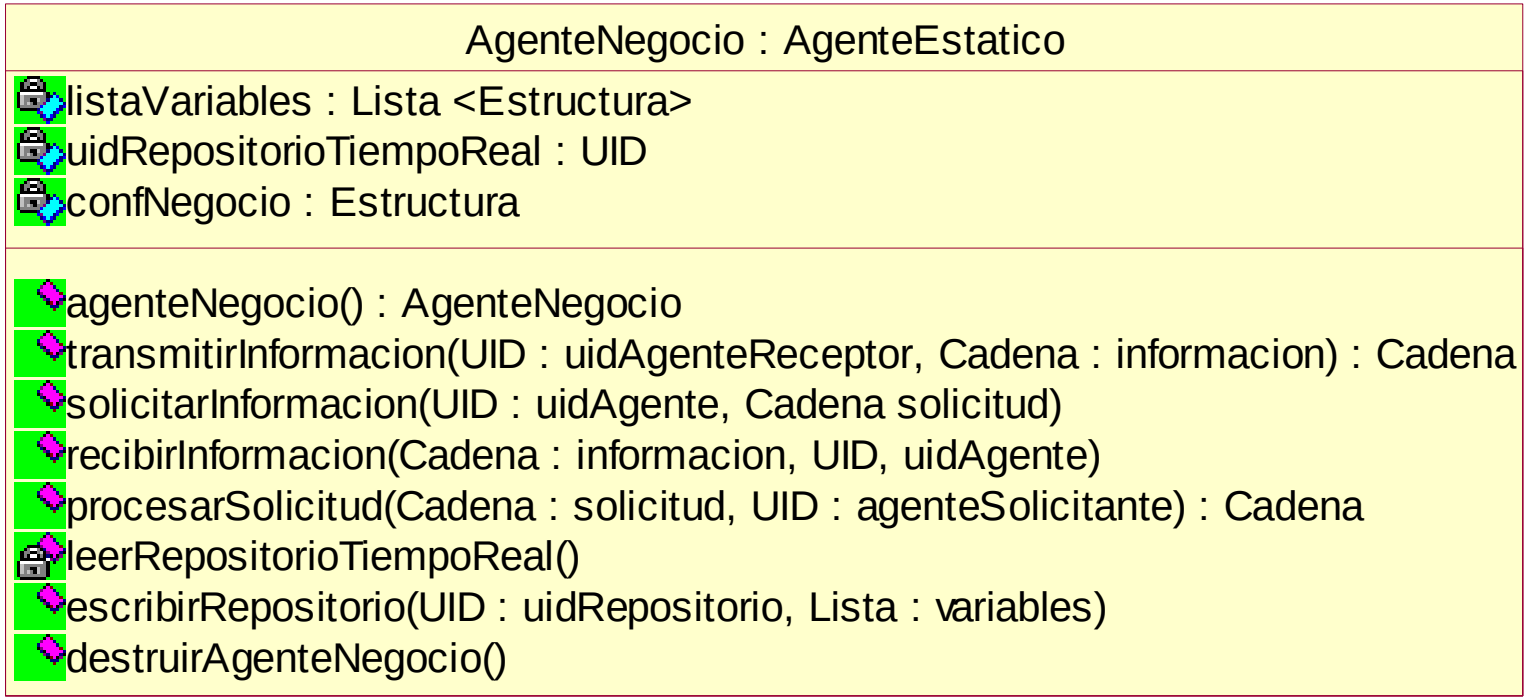
MASINA

Modelo de Diseño



UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

Diagrama de clases del Agente de Negocio



Definición del universo de clases y tipo de datos abstractos (TDAs)

20/11/07		Versión 1.0
Universo de clases y TDAs AgenteNegocio {Colección de clases y TDAs requerida para implantar el Agente de Negocio}		
1 2 3 4 5 6	Agente Negocio ( ) Cadena UID Logico TablaTiempoReal Estructura	*AgenteNegocio ( )*: clase que permite la creación de un Agente de Negocio. *Cadena*: TDA cadena de caracteres de longitud variable. *Entero*: valor entero. *UID*: tipo entero que representa un identificador único en el sistema multiagente. *Logico*: tipo lógico, conformado por los valores cierto y falso. *TablaTiempoReal:* TDA que contiene los datos del proceso real. *Estructura:* tipo de dato que contiene campos asociados a información configurada.

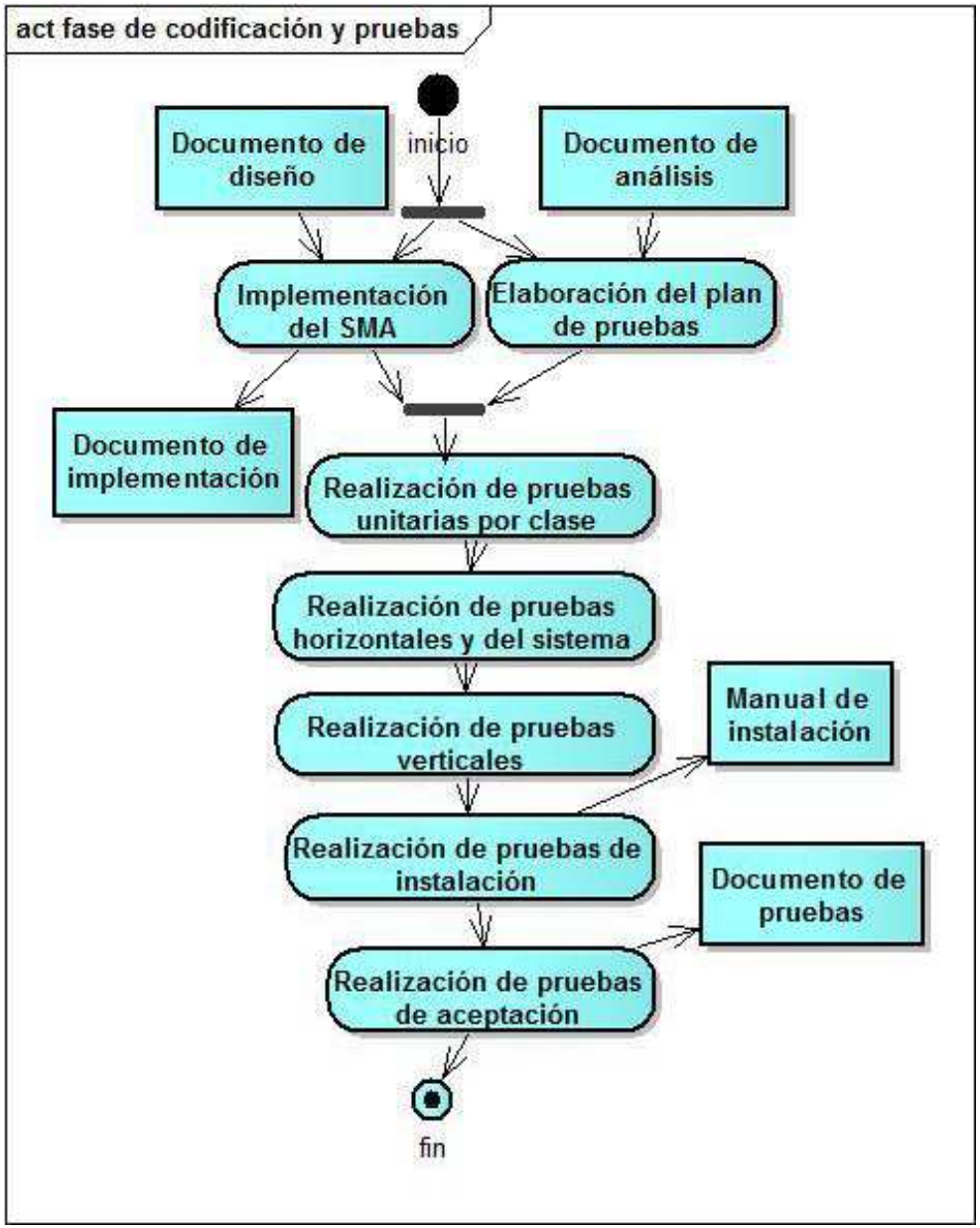
Definición del universo de clases y tipo de datos abstractos (TDAs)

20/11/07		Versión 1.0
1,1 (Constructor, Público) **agenteNegocio( ): AgenteNegocio** {Crea un Agente del tipo AgenteNegocio} {**pre:** existencia de memoria y Negocio.dat distinto de Null}		{**pos:** se crean componentes del agente o error}
1 2 3 4	abrirArchivoConf("Negocio.dat") Leer(Negocio.dat, ConfNegocio) uidRepositorioTiempoReal = asignaUid( ) listaVariables: Lista<TablaTiempoReal>	*asignaUid()*: método que asocia el AN con un único repositorio datos de tiempo real *listaVariables:* Lista que contiene los datos del proceso real
1 2	agenteNegocio x $\Rightarrow$ se creó el AN agenteNegocio x $\Rightarrow$ error	Se instancia el agente x, si éste se puede crear hay éxito, por el contrario hay error.

UNIVERSIDAD DE LOS ANDES
MÉRIDA VENEZUELA

MASINA

Fase de Codificación Y pruebas

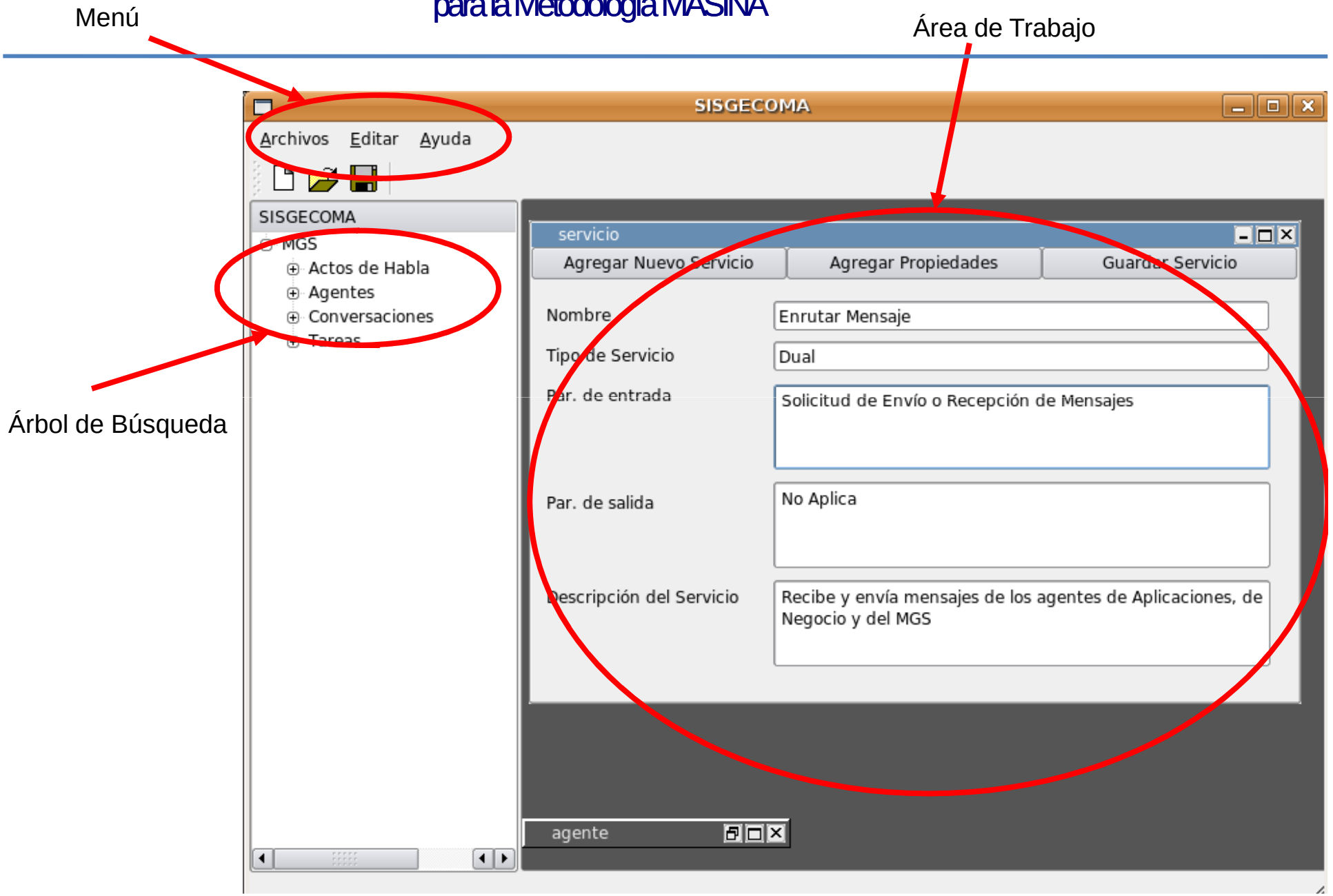


UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

MASINA

- El producto principal de esta fase consiste en un sistema de ingeniería orientado a agentes

Sistema Generador de Código para la Metodología MASINA



SISGECOMA

cemisid
centro de estudios en microelectrónica y sistemas distribuidos

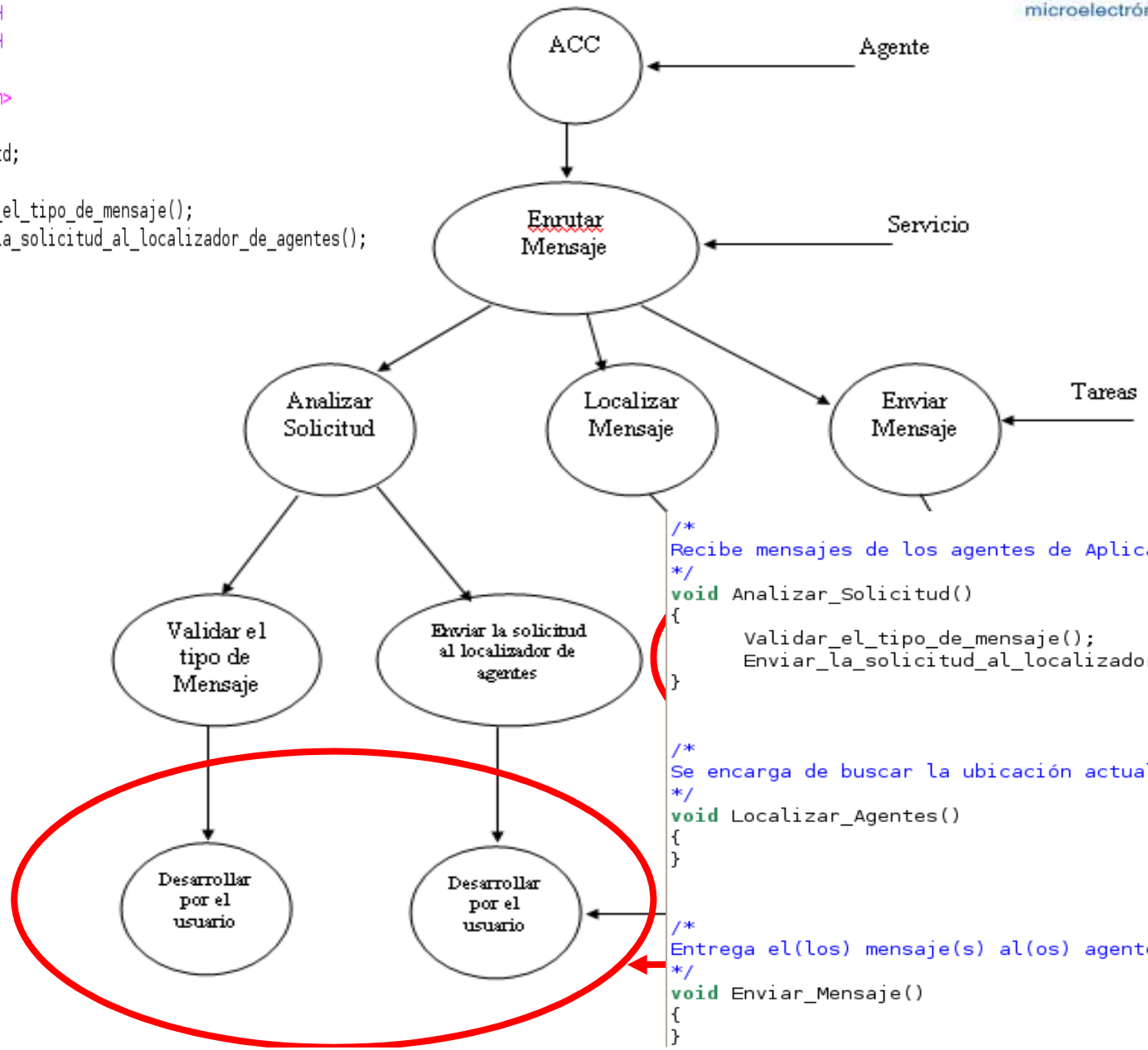
```
#ifndef SUBTAREA_H
#define SUBTAREA_H

#include <iostream>

using namespace std;

     void Validar_el_tipo_de_mensaje();
     void Enviar_la_solicitud_al_localizador_de_agentes();

#endif
```




```
/*
Recibe mensajes de los agentes de Aplicaciones, de Negoci
*/
void Analizar_Solicitud()
{
      Validar_el_tipo_de_mensaje();
      Enviar_la_solicitud_al_localizador_de_agentes();
}


/*
Se encarga de buscar la ubicación actual del(os) agente(s
*/
void Localizar_Agentes()
{
}


/*
Entrega el(los) mensaje(s) al(os) agente(s)
*/
void Enviar_Mensaje()
{
}
```

esclavo

SISGECOMA

cemisid
centro de estudios en microelectrónica y sistemas distribuidos

Servicios	Tareas
obedecer	T1 Reporte T2 Caminante

Servicios	Tareas
ordenar	T1 Manda

jefe

TAREA: Caminante	
Nombre	Caminante
Objetivo	Caminar
Descripción	Camina
Servicios asociados	obedecer
Precondición	
Sub-tareas	for(int i=1; i<11; i++) cout<<"estoy dando el paso"<<i<<endl;

```
void manda()
{
      cout<<"ponte a caminar";
}


void reporte()
{
      cout<<"hola, ya llegue"<<endl;
}


void caminante()
{
      for(int i =1; i<11;i++)
        cout<<"estoy dando el paso "<<i<<endl;
}
```

Gaia

- Una metodología software orientada al análisis y al diseño de sistemas basados en agentes.

- Su intención es la de permitir a un analista ir sistemáticamente desde los requisitos a un diseño suficientemente detallado como para implementar directamente.

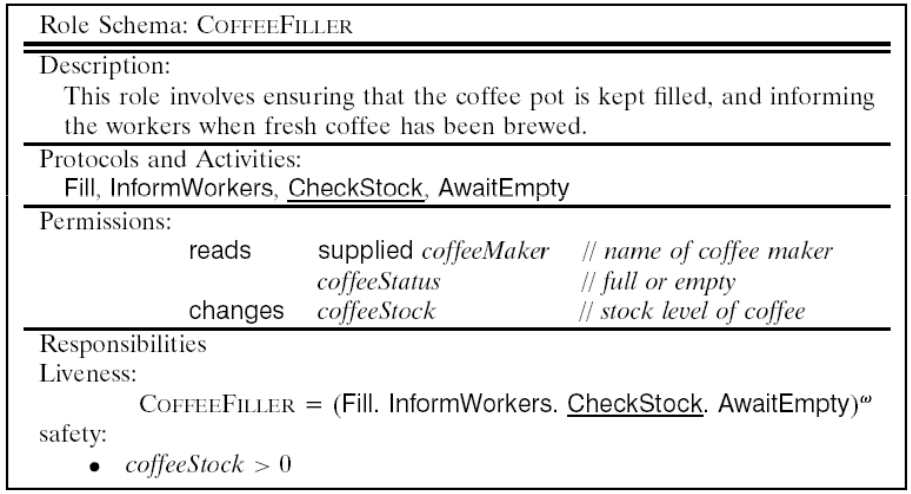
- El análisis y el diseño pueden verse como un proceso de desarrollo de modelos del sistema cada vez más detallados.

- **Organización =** una colección de roles que toman parte en patrones sistemáticos de interacción con otros roles

- **Roles =** pueden ser instancias de agentes; no fijados necesariamente

Gaia

- En el Proceso de *Análisis*:
  - Encontrar los roles en el sistema
  - Modelar las interacciones entre los roles
  - Roles consisten en 4 atributos: responsabilidades, permisos, actividades y protocolos



Role Schema: CoffeeFiller

Description:
This role involves ensuring that the coffee pot is kept filled, and informing the workers when fresh coffee has been brewed.

Protocols and Activities:
Fill, InformWorkers, CheckStock, AwaitEmpty

Permissions:

reads	supplied *coffeeMaker*	// *name of coffee maker*
	coffeeStatus	// *full or empty*
changes	*coffeeStock*	// *stock level of coffee*

Responsibilities
Liveness:
CoffeeFiller = (Fill. InformWorkers. CheckStock. AwaitEmpty)$^{\omega}$
safety:
- $coffeeStock > 0$

- En el Proceso de *Diseño*:
  - Mapear los roles en tipos de agentes
  - Crear el número correcto de instancias de agente de cada tipo
  - Modelar los servicios necesarios para desarrollar el rol
  - Crear el modelo de adquisición de conocimiento para la representación de la comunicación

Gaia

- El *análisis* incorpora los siguientes modelos:
  - **El Modelo de Rol Prototipo:** consiste en identificar los roles y dar una breve descripción de los mismos
  - **El Modelo de Interacción:** captura la interacción secuencial entre los roles
  - **El Modelo de Roles Elaborado:** documenta los roles del sistema, sus protocolos y actividades, permisos y responsabilidades
- El *diseño* incluye:
  - **El Modelo de Agente,** en el que se los roles son agregados en tipos de agentes
  - **El Modelo de Servicios,** en el que se identifican todos los servicios (funciones) asociados a cada rol de agente
  - **El Modelo de Interacción,** que simplemente define la interacción existente entre los agentes

AUML

- AUML = AGENT UNIFIED MODELING LANGUAGE (www.auml.org)
  - *"Reutilizar UML sólo donde tenga sentido"*

- UML es insuficiente para modelar sistemas multiagente
  - Comparados con los objetos, los agentes son activos ya que actúan por razones que emergen de ellos mismos

- No se trata de una metodología, sino de un lenguaje para la documentación de modelos de sistemas.

AUML

- Inicialmente se identifican dos áreas para el desarrollo detallado de especificaciones:
  - Diagramas de clases
    - Especifican el comportamiento interno de un agente y su relación con el exterior usando diagramas de clases UML adaptados
    - Las clases de UML tienen “una extensión” que son los *AgentifiedClass, AgentClass* y un AgentRoleClass
  - Diagramas de interacción
    - Término genérico que se aplica a diversos tipos de diagramas centrados en la interacción entre agentes
    - Similar a los diagramas de interacción usados en UML
    - Varios usados: Diagramas de secuencia, Diagramas de descripción de interacciones, Diagramas de colaboración, diagrama de comunicación, Diagramas de estado