



---

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA



CENTRO DE INVESTIGACIÓN EN TECNOLOGÍAS DE  
INFORMACIÓN Y SISTEMAS

*"Sistema Multiagente para la Planificación y  
Supervisión del Aprendizaje"*

Propuesta de Tesis presentada en opción del título de Maestría  
en Ciencias Computacionales

**Autor:** Euler Hernández Contreras  
[eulerconther@hotmail.com](mailto:eulerconther@hotmail.com)

**Director:** Dr. Gustavo Núñez Esquer  
[gnunez@uaeh.reduaeh.mx](mailto:gnunez@uaeh.reduaeh.mx)

Pachuca de Soto, Hgo., 15 Febrero del 2001.  
México.

## Resumen

El objetivo principal del presente trabajo es el de diseñar y realizar un sistema computacional, que sirva para planificar, guiar y verificar el proceso de aprendizaje un alumno, en un ambiente colaborativo soportado por computadora.

La investigación presenta varios aspectos, relacionados con el proceso educativo, sobre temas como: Planificación automática de Planes de Estudio, Dominio del Conocimiento, Agentes de software en el Aprendizaje Colaborativo, etc.

En éste trabajo se describirá un sistema que se ha denominado SMPSA (Sistema Multiagente para la Planificación y Supervisión del Aprendizaje).

El presente trabajo se diseñará y realizará un sistema de cómputo que permita realizar la planificación y supervisión del aprendizaje de un alumno Para lo cual se propondrá el diseño de los modelos de usuario y del dominio para poder satisfacer las necesidades del alumno.

En la presente propuesta se describen los avances que se han tenido en el desarrollo de la tesis, así como los trabajos que se deben realizar para terminarla. Lo realizado a la fecha es una propuesta, que se deberá ajustar a medida que se avance en el diseño y programación del sistema multiagente.

En el Capítulo 1 se hace una introducción al trabajo, en donde se presentan los antecedentes de la enseñanza asistida por computadora, se describen las principales aplicaciones de la computación en la educación, y se exponen los objetivos de la tesis que se elaborará.

El Capítulo 2 es el resultado de un análisis del estado del arte en los Sistemas de Aprendizaje Basados en Agentes, se presentan algunas definiciones de agentes, categorías de agentes y se hace mención de algunas aplicaciones de la tecnología de agentes en el campo educativo.

En el Capítulo 3 se presenta el Análisis del Sistema Multiagente para la Planificación y Supervisión del Aprendizaje (SMPSA).

En el capítulo 4 se presenta el Diseño del Sistema Multiagente para la Planificación y Supervisión del Aprendizaje (SMPSA).

En el capítulo 5 se explica conceptos generales del ambiente con el cual se va a trabajar para el desarrollo del sistema.

Por ultimo se menciona la bibliografía correspondiente de nuestro trabajo de investigación.

<b>1. INTRODUCCIÓN.....</b>	<b>5</b>
1.1 ANTECEDENTES.....	5
1.2 SISTEMA MULTIAGENTE PARA LA PLANIFICACIÓN Y SUPERVISIÓN DEL APRENDIZAJE.....	7
1.3 JUSTIFICACIÓN.....	8
1.4 OBJETIVOS DE LA TESIS.....	8
<b>2.SISTEMAS DE APRENDIZAJE BASADO EN AGENTES.....</b>	<b>9</b>
2.1 AGENTES DE SOFTWARE.....	9
2.2 ESTANDARIZACIÓN DE AGENTES.....	10
2.3 CATEGORÍAS DE AGENTES.....	11
2.4. SISTEMAS MULTIAGENTES.....	11
2.5 AMBIENTES DE DESARROLLO.....	13
2.5.1. <i>Agent Builder</i> .....	13
2.5.2. <i>JADE</i> .....	13
2.5.3. <i>Zeus</i> .....	13
2.5.4. <i>Madkit</i> .....	14
2.5.5. <i>JATLite</i> .....	14
2.6. PLANIFICACIÓN AUTOMÁTICA Y SCHEDULING.....	14
<b>3.ANÁLISIS DEL SISTEMA MULTIAGENTE PARA LA PLANIFICACIÓN Y SUPERVISIÓN DEL APRENDIZAJE.....</b>	<b>16</b>
3.1 ANÁLISIS.....	16
3.2. DOMINIO DEL SISTEMA MULTIAGENTE PARA LA PLANIFICACIÓN Y SUPERVISIÓN DEL APRENDIZAJE.....	16
3.3. IDENTIFICACIÓN DE DOMINIOS Y ACTORES (AGENTES).....	17
3.4. ACTORES Y PAPELES.....	18
3.6. DIAGRAMA LÓGICO DE INTERACCIÓN.....	23
3.7. DIAGRAMA DE DOMINIO DE ESTADOS.....	25
3.8. DIAGRAMA DE ESTADOS.....	26
3.8.1. <i>Agente Asistente</i> .....	27
3.8.2. <i>Agente Proveedor de Servicios</i> .....	28
3.8.3. <i>Agente Tutor</i> .....	29
3.8.4. <i>Agente Evaluador</i> .....	29
3.8.5. <i>Agente Planificador</i> .....	30

<b>4.DISEÑO DEL SISTEMA MULTIAGENTE PARA LA PLANIFICACIÓN Y SUPERVISIÓN DEL APRENDIZAJE.....</b>	<b>33</b>
4.1. DIAGRAMAS INTERNOS.....	33
4.2. DIAGRAMA DE CLASE – RELACIÓN.....	33
4.3. DIAGRAMAS DE INTERACCIÓN FÍSICA.....	34
<b>5. DESARROLLO DEL SISTEMA.....</b>	<b>36</b>
<b>6. BIBLIOGRAFÍA.....</b>	<b>37</b>

# 1. Introducción

## 1.1 Antecedentes.

Las innovaciones en la tecnología, el crecimiento y popularidad que está teniendo Internet en nuestros días, han propiciado el desarrollo de una amplia variedad de aplicaciones y proyectos de investigación en el campo educativo.

Años atrás, cuando no se tenía la tecnología de hoy en día, hubo hombres que trabajaron por hacer la enseñanza y el aprendizaje más interactivo.

En los años 20 Sydney Pressey, un psicólogo en la Universidad de Ohio, desarrolló una *"una máquina de enseñar"*, la cual proporcionaba una serie de preguntas donde el estudiante tenía que elegir la respuesta más adecuada. Esta máquina presentaba la propiedad de instruir al alumno.

F. B. Skinner fue quién sentó las bases psicológicas de la enseñanza programada, propuso que el aprendizaje se daba al modificar la conducta en el estudiante a través de un refuerzo. Skinner adoptó la máquina de enseñanza de Pressey, presentando la información en unidades llamadas marcos (frames). Después de presentar la información, se le pedía al estudiante que diera una respuesta a una pregunta, que era comparada con la respuesta correcta, si coincidía se daba el refuerzo. Skinner afirmó que el refuerzo alternado y las respuestas frecuentes modificaban la conducta del estudiante.

Norman Crowder desarrolló el programa ramificado, considerando la retro alimentación tanto en las respuestas correctas como en las incorrectas. Cuando la respuesta era correcta se sugería que se volviera a estudiar el tema donde se detectaba el error [Murray,99].

Desde los años 50, el desarrollo de las computadoras y la introducción de ella en las escuelas, se han desarrollado aplicaciones y proyectos de investigación en el campo educativo con las características de la enseñanza programada.

La Instrucción Asistida por Computadora, conocida por sus siglas en Inglés como Computer Aided Instruction (CAI), aparece heredando los métodos de enseñanza desarrollado por Skinner, la cual comprende aspectos computacionales que asisten o ayudan al proceso de enseñanza - aprendizaje.

El término de Enseñanza Asistida por Computadora (CAI) ha evolucionado, al involucrar la Inteligencia Artificial en el software educativo, denominándolo como Intelligent Computer Aided Instruction (ICAI), posteriormente este término fue reemplazado en los años 80 por el de Sistemas Tutores Inteligentes (Intelligent Tutoring System, ITS). Estos sistemas se caracterizan por reproducir el comportamiento de un tutor humano, adaptando su enseñanza - aprendizaje más conveniente para el usuario.

El desarrollo de la computación distribuida, las telecomunicaciones, las redes de computadoras, la utilización masiva de Internet y del World Wide Web han propiciado el desarrollo de sistemas que contemplan la difusión del conocimiento, la comunicación, coordinación y colaboración entre grupos de personas situados en distintos lugares geográficos., desarrollándose el Aprendizaje Cooperativo Soportado por Computadora cuyas siglas en Inglés son CSCL (Computer Supported

Collaborative Learning), desarrollado a partir de las investigaciones en el campo del Trabajo Colaborativo Soportado por Computadora (CSCW, Computer Supported Collaborative Work).

Ayala [Ayala,2000] define el Aprendizaje Colaborativo Asistido por Computadora como: *"El uso de la computadora como un dispositivo mediador que asiste a los usuarios a comunicarse, cooperar y colaborar en actividades conjuntas a través de una red, proporcionando apoyo para su coordinación en la aplicación y construcción social de conocimientos "*

Una tendencia en las tecnologías de educación cooperativa, es la de incluir un agente artificial (compañero) en un modelo de estudiante, el cual, por un lado, colabore o compita con el alumno y, por otro, que aconseje y aprenda del alumno humano.

El concepto de agentes es uno de los más importantes en los años 90, tanto en la Inteligencia Artificial Distribuida como en las Ciencias de la Informática y Computación. Se han desarrollado aplicaciones en campos tan variados como, la Administración en las telecomunicaciones, el control de tráfico aéreo, minería de datos, recuperación de la información, comercio electrónico, asistentes personales digitales, librerías digitales, control de procesos, bases de datos inteligentes y educación.

Nwana [Nwana & Ndumu, 96] define al agente como: *"Un componente de software y/o hardware, capaz de realizar una tarea en favor de un usuario"*. Shoham [Shoham, 96] define a un agente de software como: *"Una entidad de software la cual funciona continuamente y autónomamente en un ambiente en particular, a menudo habitada por otros agentes y procesos"*.

Desde el punto de vista del procesamiento distribuido se define al agente como: *"Un proceso de software autosuficiente ejecutando en forma concurrente, que encapsula algún estado y tiene la capacidad para comunicarse con otros agentes por medio de la transmisión de mensajes. "*

Un sistema basado en agentes tiene las siguientes características [Núñez et al, 99] :

- *Autonomía:* los agentes operan sin la intervención directa de los seres humanos u otros, y tienen alguna clase de control sobre sus acciones y el estado interno.
- *Capacidad Social:* los agentes interactúan con otros agentes (posiblemente seres humanos) a través de un tipo de lenguaje de comunicación entre agentes.
- *Reactividad:* los agentes perciben su ambiente (el cual puede ser el mundo físico, un usuario a través de un interfaz gráfico, la INTERNET, o todas estas combinadas) y responden de una manera oportuna frente a los cambios que ocurren en él.
- *Pro - actividad:* los agentes no simplemente actúan como respuesta a su ambiente, son capaces de exhibir un comportamiento dirigido a metas por medio de la toma de iniciativa.

Un agente es un sistema computacional que poseer las propiedades identificadas anteriormente, además se conceptualiza empleando conceptos que se aplican con más frecuencia a los seres humanos. Por ejemplo, la Inteligencia Artificial caracteriza a un agente utilizando conceptos mentales, tales como el conocimiento, la creencia, la intención y obligación.

## ***1.2 Sistema Multiagente para la Planificación y Supervisión del Aprendizaje***

El Sistema Multiagente para la Planificación y Supervisión del Aprendizaje (SMPSA), tiene como meta desarrollar de un sistema multiagente de planeación, asesoría y asistencia personal al usuario en la enseñanza - aprendizaje.

El sistema a desarrollar, se constituye de los siguientes agentes.

- Un agente Asistente (Usuario),
- Un agente Planificador,
- Un agente Tutor,
- Un agente Evaluador y
- Un agente Proveedor.

### *El agente de usuario*

Construye y mantiene el modelo de usuario representado como un conjunto de creencias acerca del usuario correspondiente.

Coopera con otros agentes compartiendo la información referente a sus modelos de usuarios, este es el interfaz entre el usuario y el ambiente de aprendizaje.

En modelo de usuario, es donde se encuentra los datos personales del usuario, así como los intereses que éste busca.

### *El agente Planificador*

Construye y mantiene el plan de aprendizaje, basándose en el modelo de usuario. Este agente esta constituido por un modelo del dominio, en donde se considera los materiales y su seriación a estudiar.

### *El agente Tutor*

Este agente se encarga de proporcionar, y de guiar al usuario de acuerdo al interés del mismo, basándose en el modelo de usuario y en el modelo del dominio.

### *El agente Evaluador*

Verifica periódicamente el avance del aprendizaje, tratando de encontrar las causas de males entendidos; se comunica con el agente planificador para reorganizar la información a estudiar, maneja el modelo del dominio y de usuario.

### *El agente Proveedor*

Es el responsable de buscar los materiales que le interesen al alumno de acuerdo a su plan de estudios.

### ***1.3 Justificación***

Son muchos los países que están utilizando las modernas tecnologías de información y comunicaciones aplicadas a la educación. Estas tecnologías de educación nos ofrecen grandes oportunidades para reacomodar el proceso de adquisición de conocimiento.

El autoaprendizaje, es una de las actividades que los seres humanos acudimos para poder desarrollar habilidades y adquirir conocimientos. Desde su aparición, las computadoras ha impactado el campo de la educación, desarrollándose una amplia variedad de aplicaciones y de proyectos de investigación, estos sistemas están enfocados a ser del tipo "tutores".

El desarrollo de los sistemas distribuidos y el avance de Internet, se ha desarrollado una explosión de información, haciendo muy importante el problema de la planificación, el guiar y supervisar el aprendizaje, así como la obtención de los materiales de interés para el usuario.

### ***1.4 Objetivos de la tesis.***

#### **Objetivo General**

Desarrollar un sistema multiagente, que permita organizar, planificar, proporcionar, guiar y evaluar el aprendizaje en base al perfil del usuario, en un ambiente colaborativo soportado por computadora.

#### **Objetivos Particulares**

- Describir y modelar los componentes del Sistema Multiagente para la Planificación y Supervisión del Aprendizaje (SMPSA).
- Diseñar e implementar los agentes que intervendrán en el sistema SMPSA.
- Diseñar e implementar el modelo de usuario, para apoyar la construcción del plan de aprendizaje.
- Diseñar e implementar el modelo del dominio, para la construcción del plan de aprendizaje.
- Desarrollar un modelo de evaluación.



## 2. Sistemas de Aprendizaje Basado en Agentes

### 2.1 Agentes de Software

La tecnología de agentes es un área de investigación de interés en nuestros días, desde la aparición de la Inteligencia Artificial Distribuida, cuya finalidad es desarrollar ambientes distribuidos inteligentes, llamados agentes, que interactúan mediante la cooperación, la competición, y la negociación.

Existen muchas definiciones del concepto de Agente. Wooldridge define la noción débil y la noción fuerte de agente como se indica a continuación [Wooldridge & Jeannings, 94]:

*Noción débil de agente:*

El término de agente es usado para denotar aquel hardware o sistema computacional que tiene las siguientes características:

- *Autonomía:* los agentes operan sin la intervención directa de los seres humanos, tienen estados internos, y tienen alguna clase de control sobre sus acciones.
- *Capacidad Social:* los agentes interactúan con otros agentes (posiblemente seres humanos), mediante un tipo de lenguaje de comunicación entre agentes.
- *Reactividad:* los agentes perciben su ambiente (el cual puede ser el mundo físico, un usuario a través de un interfaz gráfico, Internet, o todas combinadas) y responden de una manera oportuna frente a los cambios que ocurren en él.
- *Pro - actividad:* los agentes no simplemente actúan como respuesta a su ambiente, son capaces de exhibir un comportamiento dirigido a metas mediante la toma de iniciativas propias.

*Noción fuerte de agente:*

Un agente es considerado como un sistema computacional que además de tener las características antes mencionadas, tienen atributos (estados mentales) propios de los seres humanos como *conocimiento, creencias, intenciones y deseos*, entre otros.

Shoham [Shoham, 97] define aun agente como: *"una entidad de software la cual funciona continuamente y autónomamente en un ambiente en particular, a menudo habitada por otros agentes y procesos"*.

Cuando hablamos de agentes de software, existen tres dimensiones para poder medir su capacidad, las cuales son: Agencia, inteligencia y movilidad. Agencia, es el grado de autonomía y autoridad establecida en el agente; inteligencia, es el grado de razonamiento y conducta aprendida para la solución de problemas.; movilidad, es la capacidad que tienen los agentes para moverse a través de la red.

Wooldridge menciona otros atributos en el contexto de agencia. Por ejemplo [Wooldridge & Jeannings, 94]:

- *Movilidad* es la habilidad de un agente de moverse alrededor de una red electrónica.

- *Veracidad* es la suposición de que un agente no comunicará intencionalmente adrede información falsa.
- *Benevolencia* es la suposición de que los agentes no tienen metas conflictivas, y que cada agente siempre tratará de hacer lo que se le pide que haga.
- *Racionalidad* es la suposición de que un agente actuará en orden de conseguir sus metas, y no actuará de tal forma como para prevenir que sus metas sean conseguidas, al menos hasta el punto que sus creencias lo permitan.

## 2.2 Estandarización de agentes

Debido al continuo desarrollo de los agentes y su aplicación en la industria, surge la necesidad de estandarizar su implementación y asegurar su interoperabilidad continúa. Dos esfuerzos en la estandarización de agentes a son: FIPA (Foundation for Intelligent Physical Agents) y Agent Working Group, de OMG (Object Managent Group) [Bigus & Bigus,2001].

FIPA se formó en 1996 para producir normas de software para agentes heterogéneos, agentes que interactúan entre ellos y sistemas basados en agentes. Para la elaborar estas normas FIPA, requiere de la colaboración de sus miembros y del campo de agentes en general para construir especificaciones que pueden usarse para lograr interacción entre los sistemas basados en agente desarrollados por diversas compañías y organizaciones.

FIPA está organizado y estructurado en dos grupos; los que están involucrados en la producción y desarrollo de las normas; y los que soportan y/o mantienen a FIPA [2].

### *Especificaciones de FIPA*

Las especificaciones de FIPA, representan el primer paso hacia el estándar de agentes. Ellos no pretenden describir la arquitectura interna o la forma de implementar los agentes, sino que especifican la interface necesaria para soportar la interoperabilidad de los sistemas de agentes.

Cuatro áreas de estandarización han sido identificadas por FIPA, ver figura 2.1.

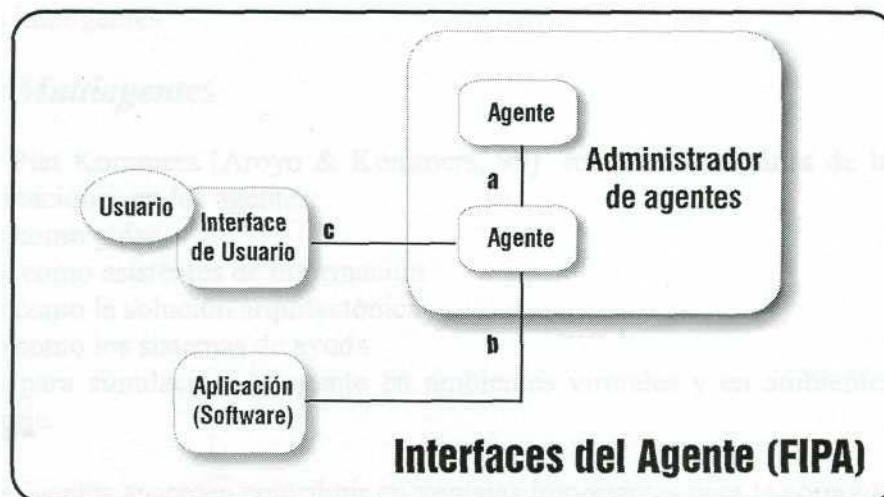


Fig. 2.1 Interfaces del Agente (FIPA)

- a) Comunicación entre agentes (a). Este facilita la comunicación entre agentes, soporta la interacción entre ellos como negociación, cooperación y el intercambio de información. Un lenguaje de comunicación entre agentes ha sido especificado por FIPA, FILA- ACL, para soportar esta interface.
- b) Administrador entre agentes. Este incluye las interfaces necesarias para soportar la creación y localización de los agentes, la comunicación entre agentes así como facilitar la movilidad y seguridad.
- c) Integración de Software (b). Esta interface soporta la integración entre un agente de software y otro que no lo es.
- d) Interacción entre agente-hombre (c). Este ve como los agentes pueden interactuar con usuarios humanos y/o otros agentes.

### **2.3 Categorías de Agentes**

Nwana propone una tipología de agentes los cuales los clasifica en [Nwana & Ndumu, 98]:

- a) Movilidad, en agentes *estáticos* o *móviles*.
- b) Por la presencia de un modelo de razonamiento simbólico, *deliberativo* o *reactivo*.
- c) Por atributos que ellos exhiben, tales como *autonomía, cooperación y aprendizaje*, derivándose tres tipos de agentes: *colaborativos, interface e inteligentes*.
- d) Por sus papeles, como *información o Internet*.
- e) *Agentes híbridos*, combinando dos o más métodos en un agente simple.
- f) Algunos agentes son clasificados de acuerdo con los *conceptos mentales*, como sus creencias, deseos e intenciones.

Nwana describe siete categorías de agentes:

1. Agentes colaborativos
2. Agentes de interface
3. Agentes móviles
4. Agentes de información / Internet
5. Agentes reactivos
6. Agentes híbridos, y
7. Agentes inteligentes

### **2.4. Sistemas Multiagentes**

Lora Aroyo y Piet Kommers [Aroyo & Kommers, 99] mencionan algunas de las tendencias del desarrollo y aplicaciones en los agentes:

- a) Agentes como guías
- b) Agentes como asistentes de información
- c) Agentes como la solución arquitectónica
- d) Agentes como los sistemas de ayuda
- e) Agentes para simulación de agente en ambientes virtuales y en ambientes interactivos de aprendizaje.

Los agentes inteligentes aparecen contribuir en ventajas importantes para la computación científica y educativa. Su aplicación en el campo educativo es principalmente el de asistente personal, guía de

usuario, sistemas de ayuda alternativos, arquitecturas de sistemas distribuidos, sistemas mediadores humanos, entre otros.

Estos cambios, implican por otro lado, que el crecimiento complejo y dinámico de las estructuras educativas necesitan ser administradas eficientemente, los nuevos tipos de servicios educativos y los mecanismos necesitan ser desarrollados y proporcionados. Esto es en particular que cada servicio necesitar satisfacer ciertos requerimientos por ejemplo, personalización, adaptación, movilidad para el usuario, apoyo para los usuarios mientras están cubriendo los nuevos tipos de tecnologías, efectividad, etc. Los agentes aparecen apoyar más eficientemente estos requerimientos en comparación con otras tecnologías existentes. Además la habilidad de ser autónomos y la cooperación entre ellos, los agentes poseen capacidad para manejar problemas como la seguridad en línea o fuera de ella y más.

La tendencia en tecnologías de educación, es la de incluir un agente artificial (compañero) en el modelo del estudiante, el cual, por un lado, colabore o compita con el alumno y, por otro, que aconseje y aprenda del alumno humano [Sheremetov & Núñez, 99].

MATHEMA, en el cual se tiene una micro-sociedad de agentes tutores artificiales que interactúan con un estudiante. Aquí se tienen dos tipos de interacciones: una entre los agentes de la micro-sociedad y otra entre el estudiante y el agente tutor.

MATHEMA fue diseñado para soportar funcionalidades y técnicas de inteligencia artificial distribuida mediante un sistema multiagente, centrándose en procesos de cooperación entre diferentes sistemas tutores, para resolver problemas complejos que no pueden ser resueltos por ningún tutor individualmente.

El sistema EduAgents de P. Hietala y T. Niemirepo [Hietala et al, 99] toma en cuenta las preferencias del alumno, esto es, el alumno lleva a cabo una serie de tareas utilizando el sistema y poco a poco va encontrando cuál es su estilo de aprendizaje. Esto se logra a través de un conjunto de agentes maestros y agentes aprendices que animan al alumno a utilizar distintas clases de aprendizaje.

En nuestro país encontramos que Universidades de renombre están utilizando la tecnología de agentes en sus proyectos, tal es el caso del Proyecto EVA (Espacios Virtuales de Aprendizaje) del Centro de Investigaciones en Computación del Instituto Politécnico Nacional; ellos han desarrollado modelos, arquitecturas y ambientes multiagentes para espacios de experimentación y aprendizaje colaborativo. Los agentes los utilizan:

- a) Como tutores virtuales, estudiantes virtuales o acompañantes de aprendizaje, asistentes virtuales personales que ayudan a los estudiantes a aprender.
- b) Para obtener información, administración, planeación y organización de las actividades de aprendizaje, filtrado y recuperación de información, asistencia al estudiante y seguimiento para la evaluación de sus intenciones y rendimiento.
- c) Para organizar los grupos de trabajo y actividades de grupo, y reconfigurar sus espacios de conocimiento y trabajo.
- d) Como componentes del espacio de experimentación de EVA, que permiten agregar el soporte de creación de Ambientes Virtuales Distribuidos en VRML/JAVA.

En la Universidad de las Américas - Puebla han desarrollado el ambiente CASSIEL (Computer Assisted Intelligent Environment for Learning) como un prototipo para ambientes de aprendizaje colaborativo enfocados a educación continua. Este sistema multi - agente permite un ambiente personalizado que establece y mantiene un plan de aprendizaje determinados por los componentes del modelo de usuario, además de la integración de grupos de discusión en la red, mediante tres agentes de software [Ayala,2000]:

- a) Un agente de usuario,
- b) Un agente de información y
- c) Un agente facilitador

## ***2.5 Ambientes de desarrollo***

En esta parte haremos mención de las herramientas que se han desarrollado para la construcción de sistemas multiagentes, algunas de estas herramientas se han comercializado, mientras que otras están en vías de desarrollo como trabajo de investigación [Bigus & Bigus,2001].

### **2.5.1. Agent Builder**

[Agent Builderl, 2002] Es un producto comercial producido por Reticular System, Inc. Este proporciona una herramienta para la construcción de agentes inteligentes, proporcionando interfaces gráficas para el diseño y desarrollo de sistemas multiagentes. Este está basado en Java y cómo lenguaje de comunicación es KQML.

Proporciona herramientas para el análisis del dominio del problema, así también de herramientas para definir la agencia (colección d e agentes inteligentes), integra y usa librerías de Java, C y C++.

### **2.5.2. JADE**

JADE (Java Agent Development Framework), es un framework implementado en Java, el cual simplifica el desarrollo de sistemas multiagentes, de acuerdo a las especificaciones que FIPA ha establecido. Proporciona un conjunto de interfaces para el desarrollo de agentes implementados en Java. JADE usa el lenguaje de comunicación de agentes de FIPA, utilizando una combinación de socket, RMIy CORBA. La última versión oficial de Jade es JADE 2.4.

### **2.5.3. Zeus**

Zeus, es una herramienta desarrollada por British Telecom., es un framework para el desarrollo de sistemas de agentes colaborativos. Zeus fue construido en java a causa de la portabilidad y el soporte de multithread.

Existen tres grupos clases en Zeus, una librería de los componentes de una agente, un conjunto de herramientas visuales, el software para construir agentes. Un agente Zeus está compuesto por tres capas: Una capa de definición, una capa organizacional y la capa de coordinación.

La capa de definición representa las capacidades del agente DBI, la capa de organización define las relaciones con otros agentes, la capa de coordinación modela cada agente como una entidad.

#### **2.5.4. Madkit**

Madkit, es una plataforma en java La cual fue construida por Oliver Gutknecht y Jacques Ferber en el LIRMM (Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier).

#### **2.5.5. JATLite**

JATLite, es un conjunto de paquetes de java que permite la construcción de agentes de software, la cual se hablará más tarde.

### ***2.6. Planificación automática y Scheduling***

El problema de planificación automática consiste en, dado un conjunto posible de acciones, un modelo del entorno, y una medida de la calidad de varios cursos de acción, generar uno o más planes de acción que satisfagan los objetivos. Scheduling es el problema más simple en el que no es necesario decidir qué acciones realizar, sino sólo cuándo. Ambos problemas tienen una importancia práctica creciente.

Thomas Dean expresa la siguiente definición de Planificación [Dean & Kambhanpati, 1996]: *"La planeación consiste en la consecuencia de actuar de acuerdo a una selección de un conjunto de posibles acciones"*

Universidades y organizaciones se han involucrado en la Planificación y Scheduling creando grupos de investigación como el Planning and Scheduling Special Interest Group (PlanSIG) en el Reino Unido, interesados en los alcances de la Inteligencia Artificial en la Planificación y Scheduling así como sus aplicaciones.

Algunas instituciones o universidades involucradas en ello son:

La Escuela de Ciencias de la Computación de la Universidad de Carnegie Mellon, desarrolló el Sistema Scheduling DTOPS, una herramienta para la generación, análisis y revisión de schedules de acciones logísticas. Este fue desarrollado a causa de la falta de herramientas para el horario de transportes.

Graphplan es un planificador de propósito general para dominios STRIPS- estilo, basados en algoritmos de grafos. Graphplan en base a un problema dado, él construye una estructura compacta llamada Grafo de Planificación, en el que un plan es un tipo de "flujo" de valores verdaderos a través del gráfico. Este grafo tiene la propiedad de tomar la información útil y puede ser propagado rápidamente a través del grafo cuando se está construyendo. Graphplan toma esta información en la búsqueda de un plan. Graphplan fue creado por Avrim Blum y Merrick Furst, con las extensiones

subsecuentes y mejoras hechas por investigadores de diferentes instituciones del mundo [Blum & Furst, 1997].

ChariTime es un proyecto de investigación enfocado al diseño, desarrollo y evaluación de un sistema distribuido para administrar y organizar citas médicas. ChariTime esta basado en la programación orientada a agentes. Aunque el principal enfoque miente sobre las citas médicas, los conceptos ChariTime son flexibles para ser aplicados en cualquier red de proveedores y requerimientos de servicios. Este proyecto es llevado a cabo en el Centro de Tecnologías de Información en Alemania [Hannebauer & Geske, 1999].

David E. Wilkins, desarrollo una arquitectura multiagente planificador, el cual es un frame que integra diversas tecnologías para resolver problemas de planificación.

Sistema de Control Escolar GES, cuya finalidad es automatizar los procesos de atención de cualquier institución educativa, incluyendo la generación de horarios y programación de los cursos.

Algoritmos Genéticos son usados para optimizar funciones que mejoran la efectividad en la producción de esquemas de planificación y scheduling.

Jürgen Dorn, desarrolló una procedimiento para comunicar sistemas expertos scheduling basado en lógica difusa, teniendo ventaja de que más información puede ser transmitida en casos de comunicación de datos precisos [Dorn & Kerr, 199].

Los agentes por las cacterísticas que poseen se han introducido en la Planificación y Scheduling, en el siguiente cuadro veremos cuatro ejemplos de sistemas de planificación y scheduling.

Sistema	¿Qué es un agente?	Terminología
AARIA (Autonomous Agents for Rock Island Arsenal)	Fuentes (por ejemplo: maquinas, y operadores), las partes, administración de procesos, partes componentes entre la unidad de proceso y su recurso	Agente coordinador de manejo; scheduling de recursos.
Daewoo	Tarea de agentes (ordenadores de trabajo), agentes fuentes (maquinas), agentes servidores	Coordinación; generación de herencia hacia las comunidades de agentes.
ISCM (Integrated Suply Chain Management)	Agentes funcionales: Logística, Adquisición de orden y agentes de transporte	Inteligencia Artificial tradicional sin agentes funcionales.
LMS (Logistics Management System)	Una critica para cada característica de la producción	Combina el juicio de votos desde criticas basadas en su bjetivo individual.

### 3. Análisis del Sistema Multiagente para la Planificación y Supervisión del aprendizaje.

#### **3.1 Análisis**

La metodología del desarrollo de *software* orientado a agentes forma parte de una metodología integrada por el desarrollo orientado a procesos, orientado a datos, orientado a objetos y orientado a agentes que han sido desarrollados durante estos últimos años[Pont,97].

El análisis orientado a agentes consta de los siguientes pasos:

- Los agentes externos (identificación del sistema), se identifican los agentes externos, y se destacan sus metas.  
Se considera quién utilizará el sistema, con qué otros sistemas deben obrar recíprocamente y qué dotación física nuestro sistema obrará recíprocamente.
- Los diagramas lógicos de la interacción (conversación), se crea un diagrama de interacciones, incorporando un "walkthrough".  
Los diagramas de interacción [Booch, 94] proporcionan una manera natural de representar el control y el tiempo de la información; los walkthroughs son una manera extensa y rentable de repasar y refinar los requisitos del sistema [Norris et al, 93].
- El sistema y los mensajes externos, se identifican los mensajes a los cuales el sistema debe responder.
- El diagrama del dominio del mensaje; cuando creamos el diagrama del dominio del mensaje, estamos terminando nuestro resumen de los que hace la necesidad del sistema, de la perspectiva de la gente y de otros agentes con quienes debe interactuar.

#### **3.2. Dominio del Sistema Multiagente para la Planificación y Supervisión del Aprendizaje**

El dominio del ambiente de aprendizaje basado en agentes está modelado en términos de agentes, representado las necesidades del usuario y las políticas de negocio.

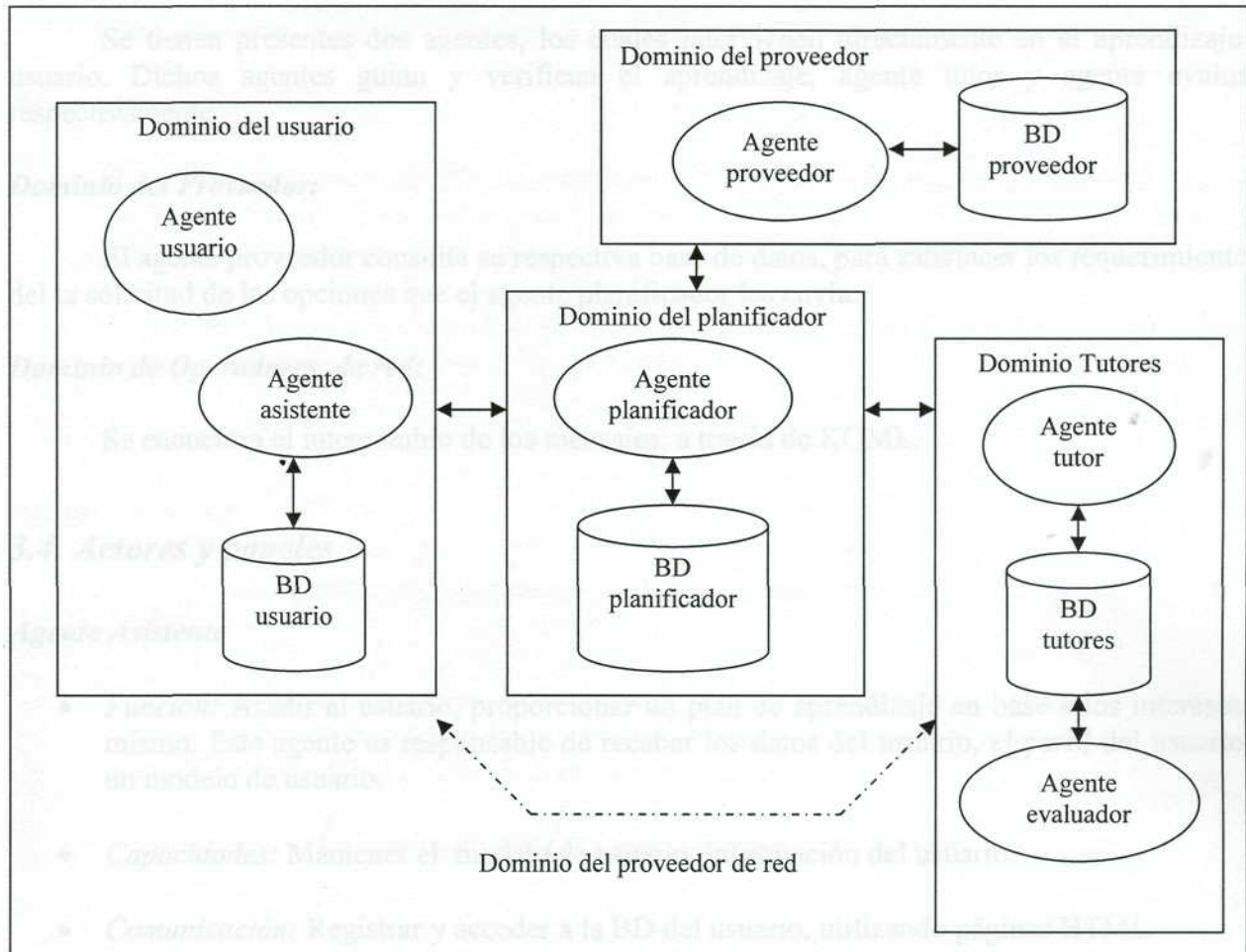
Escenarios generales:

Interacción típico de un usuario y el ambiente de aprendizaje basado en agentes.

1. El usuario le dice al agente Asistente que le propongan un plan para poder estudiar un tema.
2. El agente asistente le pide datos al usuario para poder registrarlo, en caso que éste se encuentre registrado, el agente extrae de su base de datos el perfil del usuario.
3. El agente planificador elabora un plan de aprendizaje de acuerdo al perfil del usuario.
4. El agente planificador registra aquellas instancias que el proveen información para el usuario.
5. El agente planificador envía los planes de estudios a los agentes tutor y evaluador.



6. El agente tutor recibe la información proporcionada por el agente facilitador (y éste del agente proveedor), para poder guiar al usuario de acuerdo al plan proporcionado por el agente facilitador.
7. El agente evaluador, evalúa el aprendizaje del usuario.
8. Si el usuario está de acuerdo a los temas que le son proporcionados por el agente asistente; entonces el SMPSA se compromete a cumplir lo especificado.



### 3.3. Identificación de dominios y actores (agentes).

#### **Dominio de usuario:**

Los usuarios desean estudiar temas de su interés que proporciona el SMPSA (Sistema Multiagente para la Planificación y Supervisión del Aprendizaje). El agente asistente, confirma la solicitud que el usuario hace para poder entrar al SMPSA.

### ***Dominio del planificador:***

Se registran los diferentes institutos participantes para la solicitud del servicio, el agente planificador es el encargado de realizar este registro. Es aquí donde se elabora el plan de aprendizaje en base al modelo de usuario y del dominio, el cual será proporcionado a los demás agentes.

### ***Dominio de los Tutores:***

Se tienen presentes dos agentes, los cuales intervienen directamente en el aprendizaje del usuario. Dichos agentes guían y verifican el aprendizaje, agente tutor y agente evaluador respectivamente.

### ***Dominio del Proveedor:***

El agente proveedor consulta su respectiva base de datos, para satisfacer los requerimientos de la solicitud de las opciones que el agente planificador les envía.

### ***Dominio de Operadores de red:***

Se encuentra el intercambio de los mensajes, a través de KQML.

## ***3.4. Actores y papeles***

### ***Agente Asistente***

- *Función:* Asistir al usuario, proporcionar un plan de aprendizaje en base a los intereses del mismo. Este agente es responsable de recabar los datos del usuario, el perfil del usuario, en un modelo de usuario.
- *Capacidades:* Mantener el modelo de usuario, información del usuario.
- *Comunicación:* Registrar y acceder a la BD del usuario, utilizando páginas HTML.

### ***Agente Planificador***

- *Función:* Define los planes de estudios de acuerdo al modelo de usuario y de su dominio. Registra los institutos participantes para el desarrollo del plan de aprendizaje. Se relaciona con el agente tutor para proporcionar el material que se va a estudiar; y con el agente evaluador para que este le proporcione la evaluación y así enviarla al agente asistente.
- *Capacidades:* Envía el plan de estudios al agente tutor y evaluador. Registra la respuesta por parte de los agentes tutor y evaluador. Se comunica con el agente proveedor de servicios para consultar la información que ellos contienen.
- *Comunicación:* Se comunica con los agentes a través de KQML.

### ***Agente Tutor***

- *Función:* Este agente está a cargo de proporcionar, y de guiar al usuario el material de su interés.
- *Capacidad:* Recibe la información del material que los proveedores dan al agente planificador. Toma y registra en la BD Tutores el avance que ha tenido con el usuario.
- *Comunicación:* A través de KQML con otros agentes.

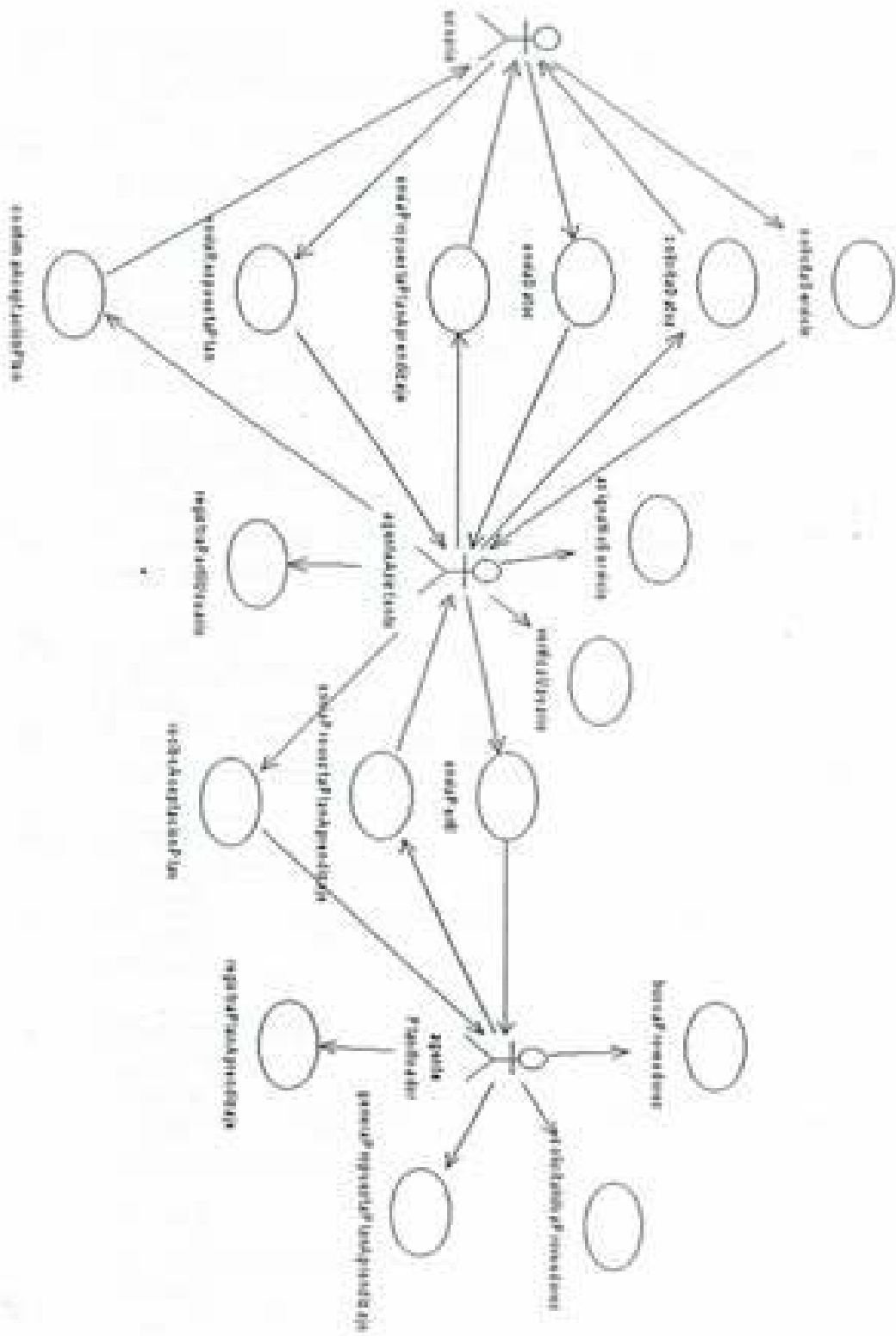
### ***Agente Evaluador***

- *Función:* Este agente verifica periódicamente el avance de aprendizaje, tratando de encontrar las causas de males entendidos; se comunica con el agente facilitador para reorganizar la información a estudiar.
- *Capacidad:* Toma y registra en la BD Tutores las calificaciones del usuario, también aquellos procesos que le permiten la evaluar al usuario.
- *Comunicación:* Es a través de KQML con otros agentes.

### ***Agente Proveedor***

- *Función:* Es el responsable de proporcionar al agente planificador los diferentes líneas de estudio.
- *Capacidad:* Toma de su base de datos, la información solicitada por el agente planificador, dándole a conocer los diferentes opciones a estudiar. Proporciona al agente planificador el material de interés que el usuario pretende estudiar.
- *Comunicación:* A través de KQML con otros agentes.

3.4.1 Diagrama de Casos de Uso.



### 3.5. Documentación de los caso de uso.

<b>Caso de Uso:</b>	SolicitaServicio
<b>Actores:</b>	Usuario, agenteAsistente
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El usuario solicitar el servicio para poder suplir su necesidad de aprender un tema correspondiendo a sus deseos.
<b>Caso de Uso:</b>	asignaNoServicio
<b>Actores:</b>	agenteAsistente
<b>Tipo:</b>	Primario
<b>Descripción:</b>	Se tiene como propósito que el agente asistente asigne un número de servicio para llevar el control de usuarios que entran al sistema.
<b>Caso de Uso:</b>	verificaUsuario
<b>Actores:</b>	agenteAsistente
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente asistente verifica si el usuario es nuevo o si desea entrar al sistema, validando la cuenta y el password del usuario.
<b>Caso de Uso:</b>	solicitaDatos
<b>Actores:</b>	Usuario, agenteAsistente
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente asistente para poder ofrecer el servicio, tiene que registrar al usuario, solicitando datos del usuario, así como los intereses del mismo.
<b>Caso de Uso:</b>	enviaDatos
<b>Actores:</b>	Usuario, agenteAsistente
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El usuario envía sus datos correspondientes para obtener el servicio solicitado.
<b>Caso de Uso:</b>	registraPerfilUsuario
<b>Actores:</b>	agenteAsistente
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente lleva a cabo el registro de los datos del usuario, así como los intereses del mismo.
<b>Caso de Uso:</b>	envíaPerfil
<b>Actores:</b>	AgenteAsistente, agentePlanificador
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente asistente envía el perfil del usuario para que en base a ello lleve a cabo la generación del plan de aprendizaje basado a los intereses del usuario.

<b>Caso de Uso:</b>	BuscaProveedores
<b>Actores:</b>	agentePlanificador
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente planificador consulta los proveedores (los materiales) que tiene disponible para general el plan de aprendizaje.

<b>Caso de Uso:</b>	solicitaInfoaProveedores
<b>Actores:</b>	AgentePlanificador
<b>Tipo:</b>	Primario
<b>Descripción:</b>	En este caso de uso el agente planificador se encargara de obtener información de los proveedores, es decir las programas de las materias disponibles, para general el plan de aprendizaje.

<b>Caso de Uso:</b>	generaPropuestaPlanAprendizaje
<b>Actores:</b>	agentePlanificador
<b>Tipo:</b>	Primario
<b>Descripción:</b>	Es en este caso de uso donde se genera el plan de aprendizaje.

<b>Caso de Uso:</b>	enviaPropuestaPlanAprendizaje
<b>Actores:</b>	AgentePlanificador, agenteAsistente
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente planificador envía el plan de aprendizaje generado por él al agente asistente.

<b>Caso de Uso:</b>	enviaPropuestaPlanAprendizaje
<b>Actores:</b>	AgenteAsistente, usuario
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente asistente recibe el plan de aprendizaje y lo envía al usuario.

<b>Caso de Uso:</b>	enviaRespuestaPlan
<b>Actores:</b>	Usuario, agenteAsistente
<b>Tipo:</b>	Primario
<b>Descripción:</b>	En este caso de uso el usuario, envía al agente asistente la aceptación o rechazo del el plan de aprendizaje generado por el sistema.

<b>Caso de Uso:</b>	recibeAceptacionPlan
<b>Actores:</b>	agenteAsistente, agentePlanificador
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente planificador recibe aceptación del el plan de aprendizaje generado por él.

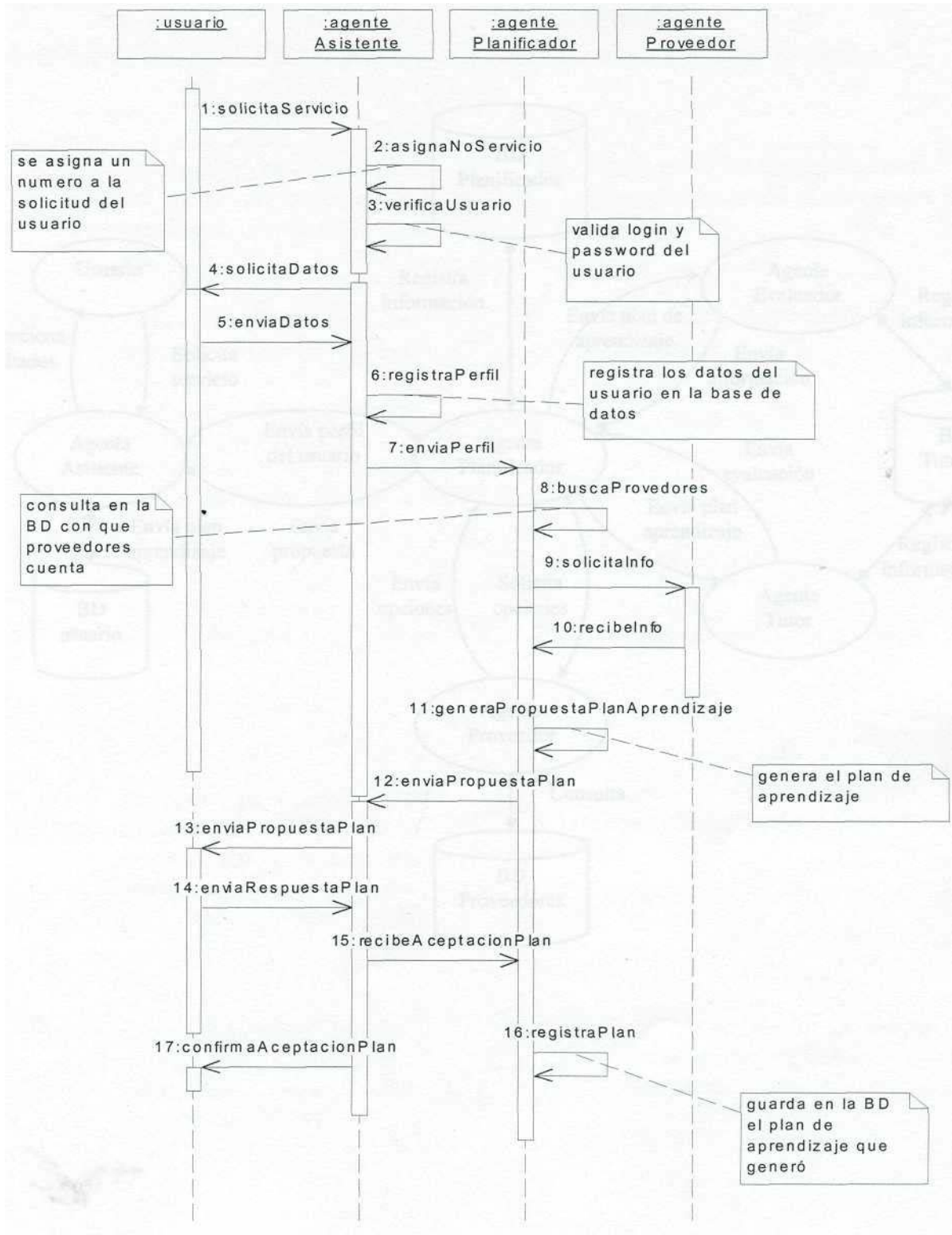
<b>Caso de Uso:</b>	registraPlanAprendizaje
<b>Actores:</b>	agentePlanificador
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente planificador registra el plan de aprendizaje para que en base a ello, se lleve a cabo la enseñanza de los temas y consultarlo para cuando el usuario desee entrar a otra sesión.
<b>Caso de Uso:</b>	confirmaAceptacionPlan
<b>Actores:</b>	AgenteAsistente, usuario
<b>Tipo:</b>	Primario
<b>Descripción:</b>	El agente asistente confirma que el plan de aprendizaje esta registrado; además le informa que en cualquier momento el usuario puede entrar al sistema a estudiar los temas establecidos en el plan.

### ***3.6. Diagrama lógico de interacción.***

Este diagrama muestra el escenario del sistema, mostrando el turno y comportamiento de los mensajes.

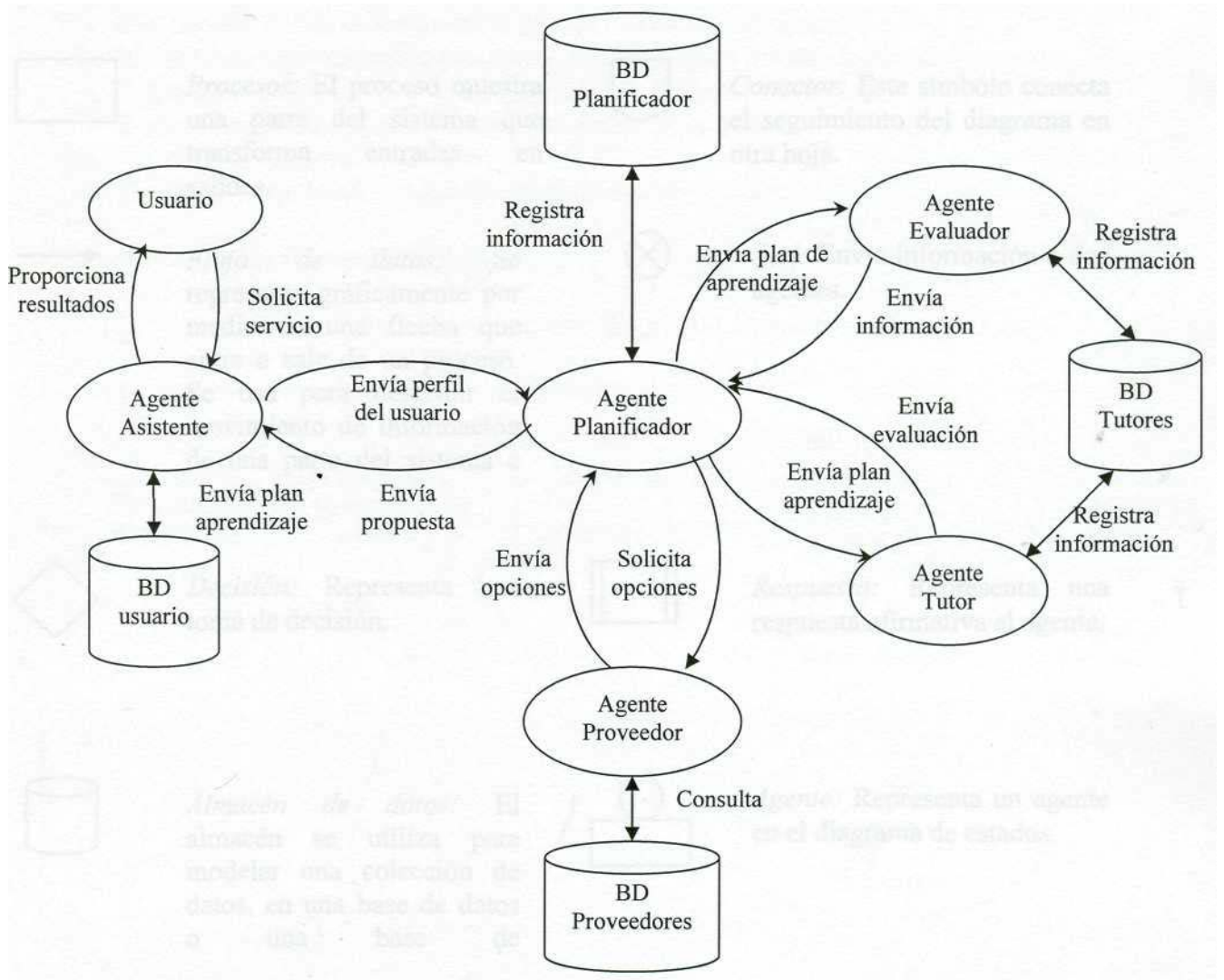
- 1) El usuario solicita que le den información de que temas él puede estudiar.
- 2) El agente asistente se encarga de recolectar el perfil del usuario, almacenado los datos del usuario.
- 3) El agente asistente envía el perfil del usuario al agente planificador.
- 4) El agente planificador (el intermediador entre los agentes asistente, proveedor y los tutores), registra los agentes proveedores de servicios en su base de datos.
- 5) El agente planificador elabora y envía el plan de aprendizaje de acuerdo al perfil del usuario, solicitando las opciones que le ofrece el agente proveedor de servicios.
- 6) El agente proveedor de servicios busca la información solicitada, consultando su base de datos, los cuales contienen los temas a estudiar.
- 7) El agente proveedor de servicios envía las posibles opciones que encontró en su base de datos al agente planificador.
- 8) El agente planificador recopila la información que recibe del agente proveedor de servicios, enviando la propuesta al agente asistente.
- 9) El agente asistente recibe la información registrándola y enviándola al usuario.
- 10) El agente asistente espera respuesta del usuario.
- 11) El usuario decide una opción.
- 12) El agente asistente registra la respuesta y la envía al agente planificador.
- 13) El agente planificador registra y compromete al proveedor de servicios que le proporcione la información correspondiente.
- 14) El agente proveedor de servicios proporciona la información al agente planificador.
- 15) El agente planificador envía el material de acuerdo al perfil del usuario.
- 16) Los agentes tutores guían y verifican el aprendizaje del usuario.
- 17) Los agentes tutores envía resultados del aprendizaje al agente asistente.
- 18) El agente asistente entrega resultados.
- 19) El agente asistente termina sesión.

### Diagrama lógico de interacción.





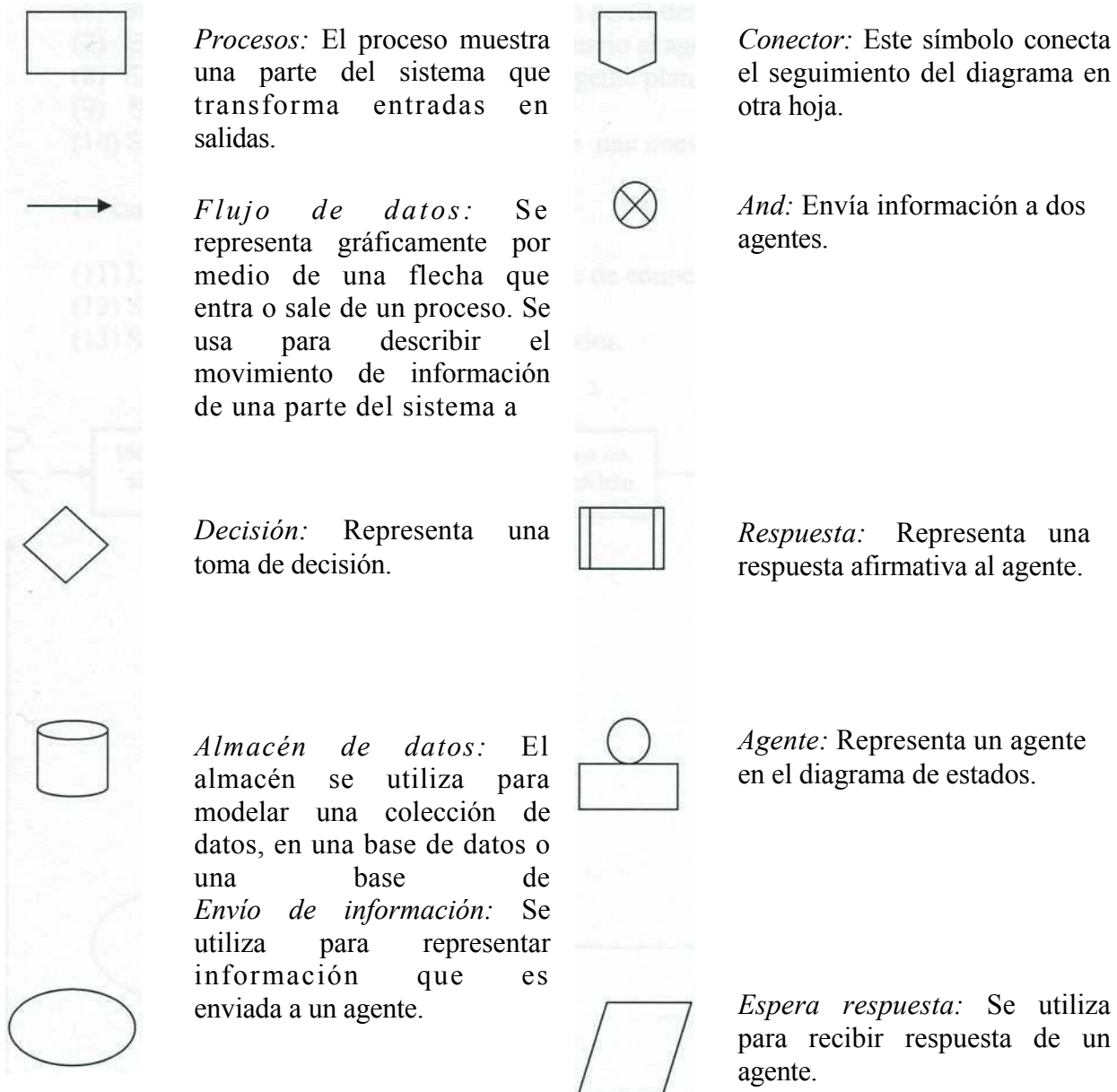
### 3.7. Diagrama de Dominio de estados.



### 3.8. Diagrama de Estados.

El diagrama de estados muestra los diferentes sucesos que se realizan en el sistema multiagente.

Antes de iniciar, indicaremos la notación que se utilizaremos en los diagramas de estados.

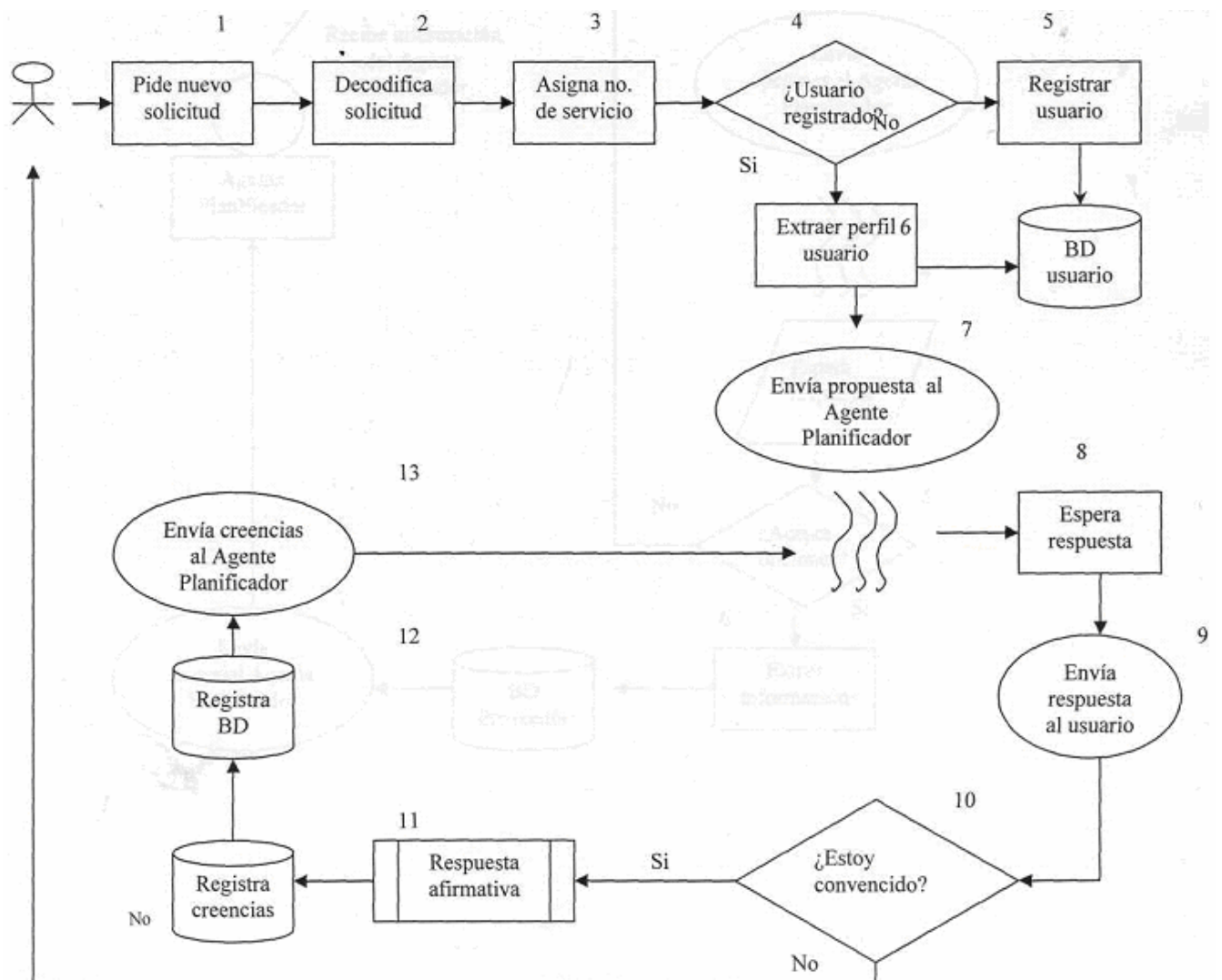


### 3.8.1. Agente Asistente

- (1) El agente asistente espera una solicitud por parte del usuario.
- (2) Procede a decodificar la solicitud.
- (3) Se asigna un número de servicio.
- (4) Pregunta si el usuario está registrado.
- (5) Si el usuario es nuevo, entonces se procede a registrar al usuario.
- (6) Si el usuario está registrado extrae su perfil desde la base de datos.
- (7) El agente asistente envía los datos del usuario al agente planificador.
- (8) El agente asistente espera respuesta del agente planificador.
- (9) Se envía la respuesta al usuario.
- (10) Si el usuario no está convencido, se pide una nueva solicitud.

En caso de ser una respuesta afirmativa:

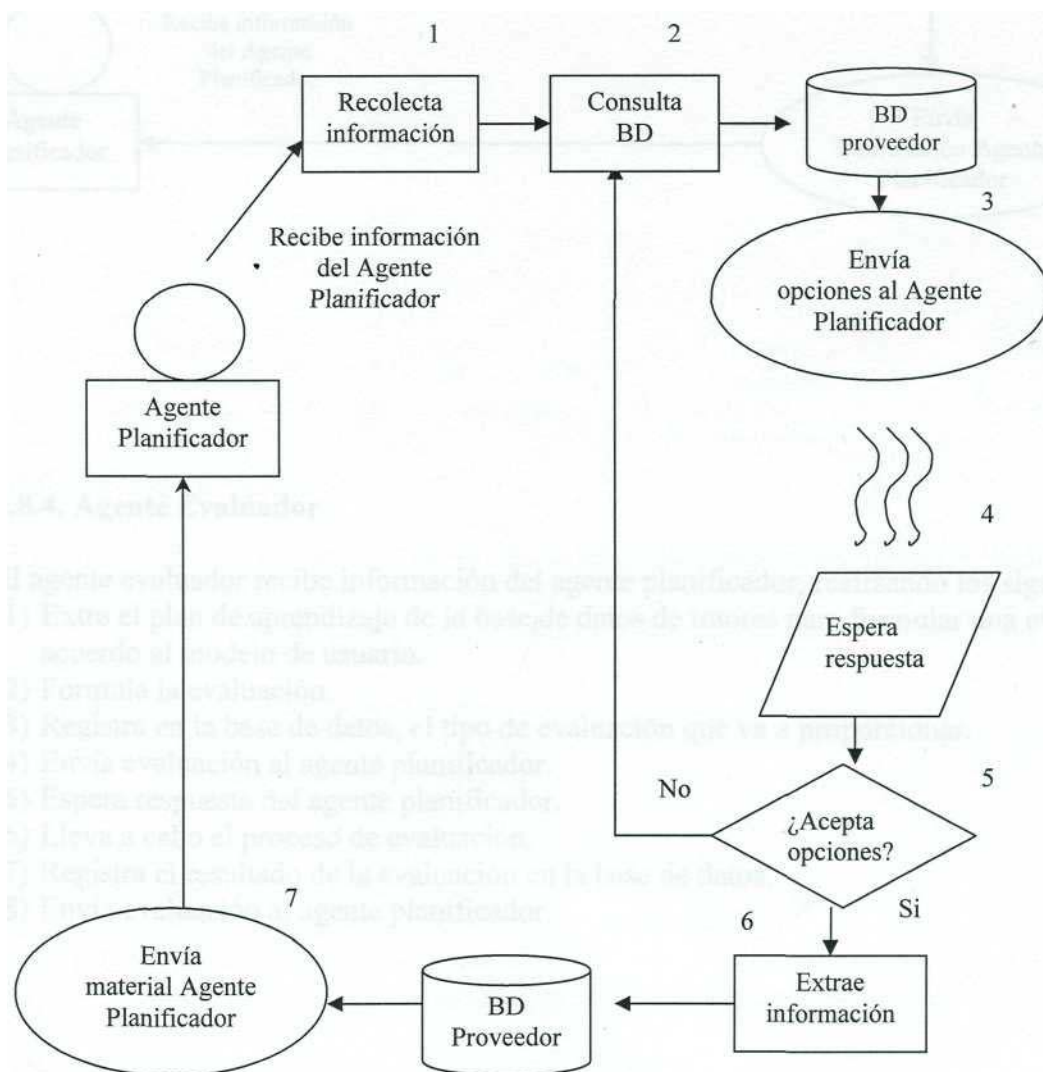
- (11) Las creencias son registradas en una base de conocimientos del agente asistente.
- (12) Se registra en la base de datos.
- (13) Se envía las creencias al agente planificador.



### 3.8.2. Agente Proveedor de Servicios

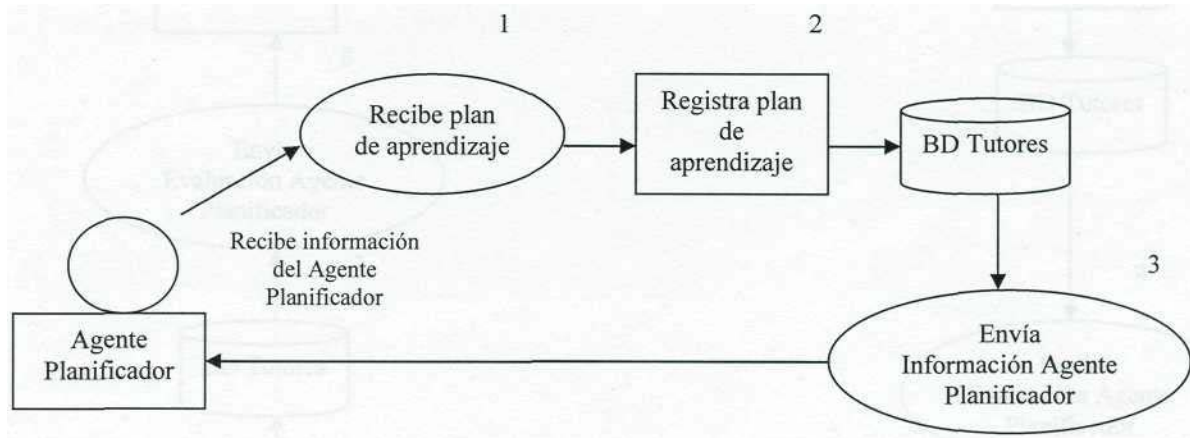
El agente proveedor de servicios realiza lo siguiente:

- (1) El agente proveedor de servicios recibe información del agente planificador para su consulta.
- (2) Consulta su base de datos de sus proveedores.
- (3) Envía las opciones al agente planificador.
- (4) Espera respuesta del agente planificador.
- (5) Si el usuario no acepta las opciones, realiza una nueva consulta.
- (6) Si el usuario acepta las opciones, extrae la información de su base de datos.
- (7) La información es enviada al agente planificador.



### 3.8.3. Agente Tutor

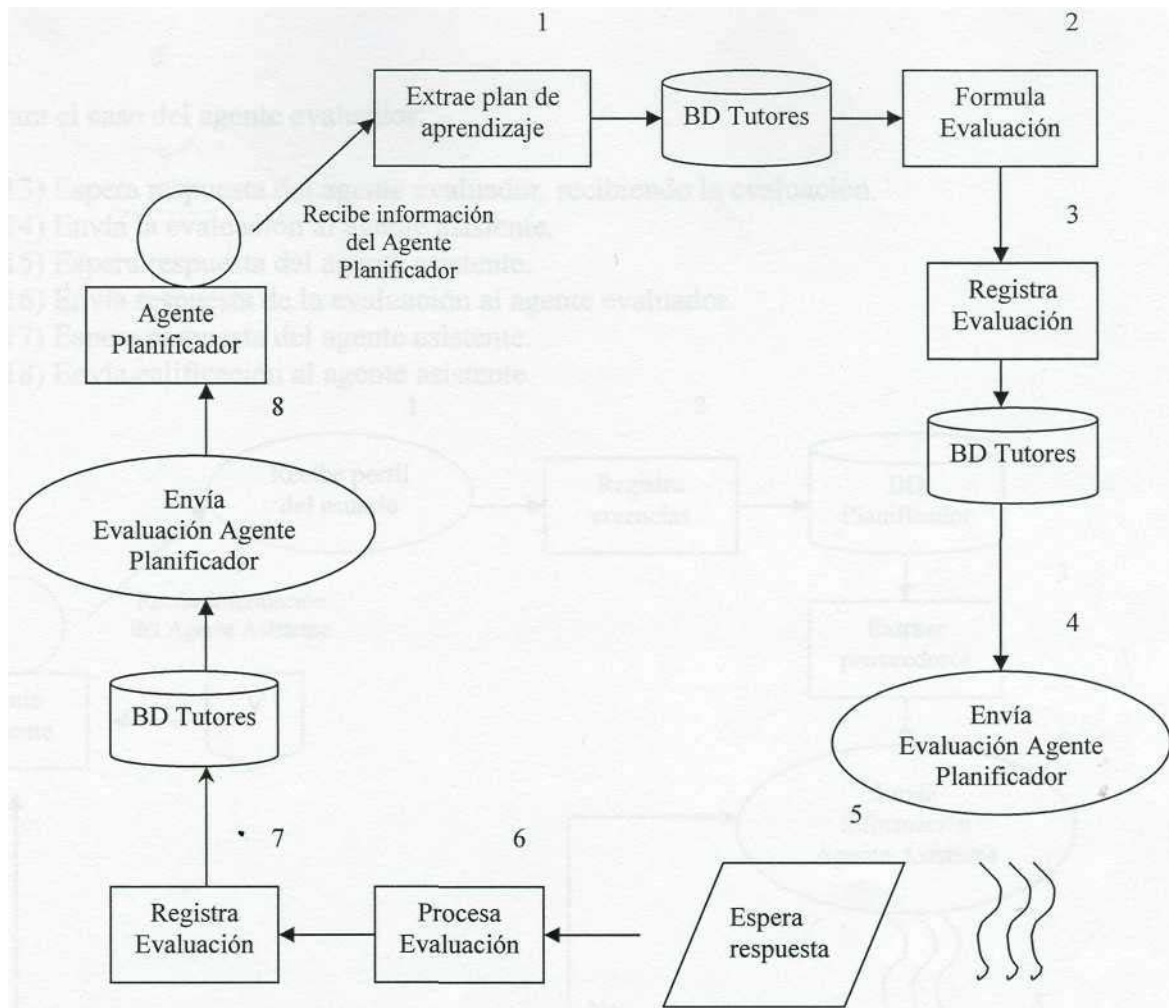
- (1) El agente tutor recibe el plan de aprendizaje
- (2) Registra el plan de aprendizaje en la base de datos .
- (3) Envía la información al agente planificador.



### 3.8.4. Agente Evaluador

El agente evaluador recibe información del agente planificador, realizando los siguiente:

- (1) Extra el plan de aprendizaje de la base de datos de tutores para formular una evaluación de acuerdo al modelo de usuario.
- (2) Formula la evaluación.
- (3) Registra en la base de datos, el tipo de evaluación que va a proporcionar.
- (4) Envía evaluación al agente planificador.
- (5) Espera respuesta del agente planificador.
- (6) Lleva a cabo el proceso de evaluación.
- (7) Registra el resultado de la evaluación en la base de datos.
- (8) Envía evaluación al agente planificador.



### 3.8.5. Agente Planificador

El agente planificador es el que registra los agentes proveedores de servicios para posteriormente solicitarle las peticiones que le son enviadas.

- (1) El agente planificador recibe las solicitudes enviadas por el agente asistente (recibe el perfil del usuario).
- (2) Estas son registradas como creencias.
- (3) Extrae de su base de datos los proveedores en los cuales el puede recibir el servicio.
- (4) Envía la información al agente asistente.
- (5) Espera respuesta del agente asistente.
- (6) Si el usuario no está convencido, al agente asistente.
- (7) Si está convencido registra la solicitud como creencia.
- (8) Pide que le proveen información al agente proveedor.
- (9) Espera respuesta del agente proveedor.
- (10) La información es enviada a los agentes tutor y evaluador.

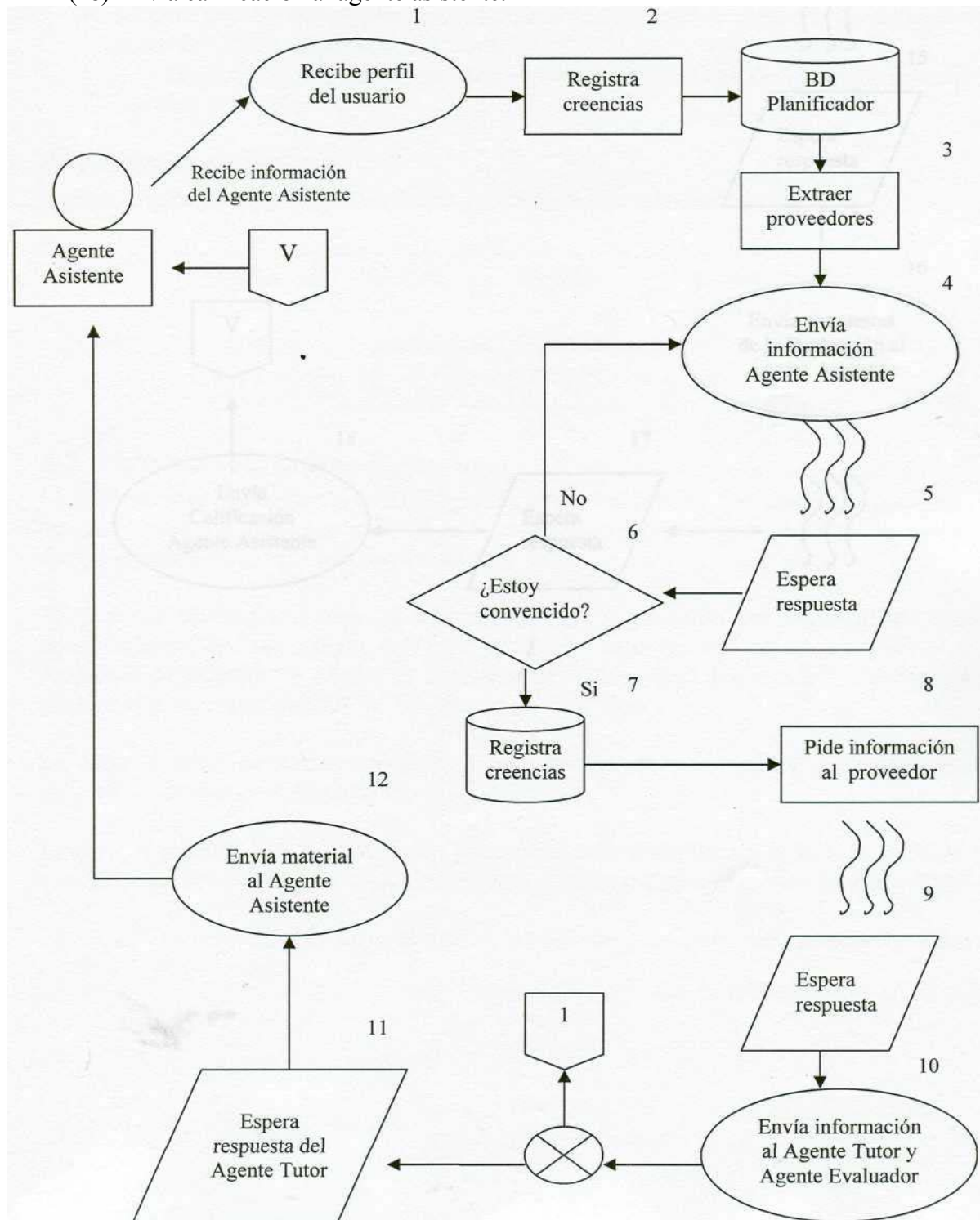
Para el caso del agente tutor:

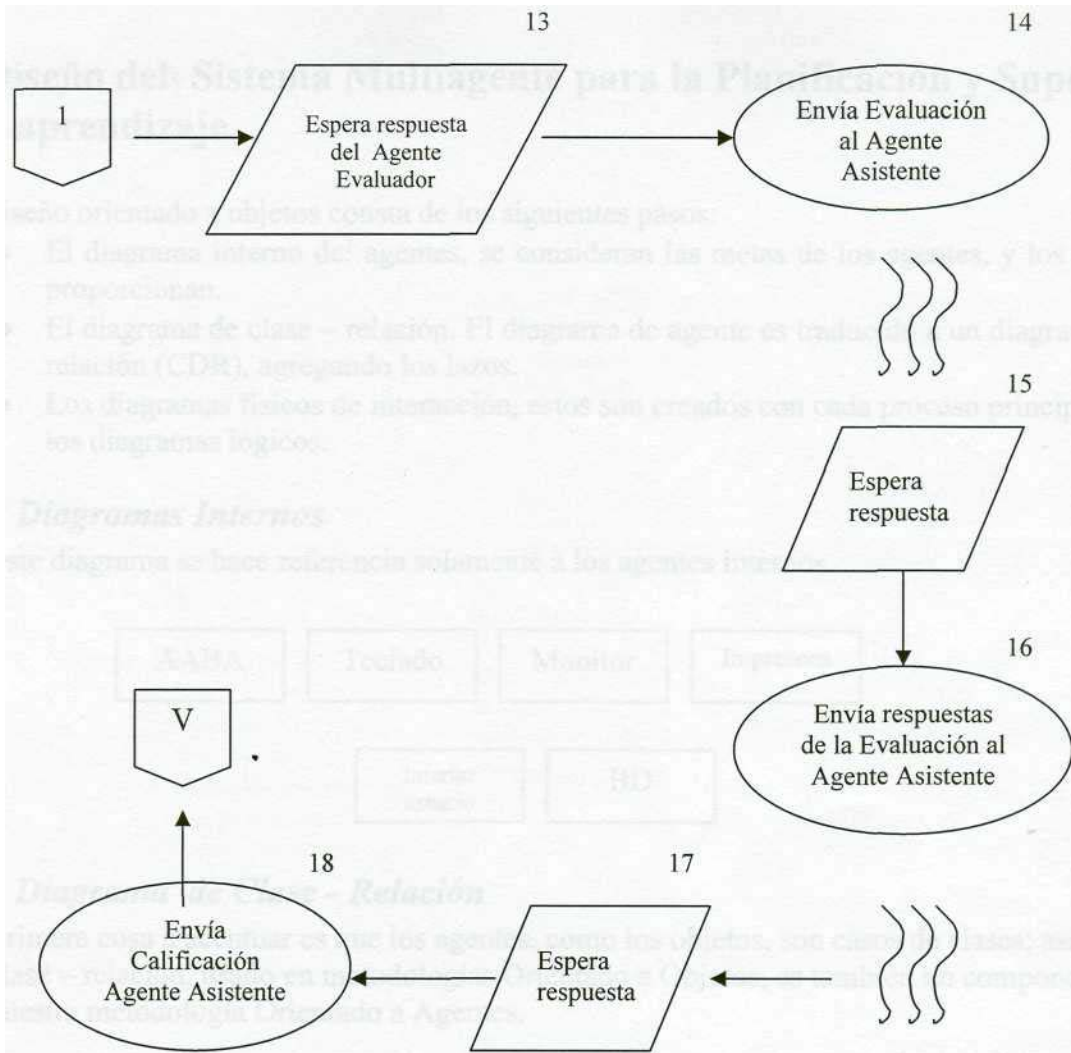
- (11) Espera la respuesta del agente tutor.
- (12) Envía material al agente asistente.



Para el caso del agente evaluador:

- (13) Espera respuesta del agente evaluador, recibiendo la evaluación.
- (14) Envía la evaluación al agente asistente.
- (15) Espera respuesta del agente asistente.
- (16) Envía respuesta de la evaluación al agente evaluador.
- (17) Espera respuesta del agente asistente.
- (18) Envía calificación al agente asistente.







## 4. Diseño del Sistema Multiagente para la Planificación y Supervisión del aprendizaje.

El diseño orientado a objetos consta de los siguientes pasos:

- El diagrama interno del agentes, se consideran las metas de los agentes, y los servicios que proporcionan.
- El diagrama de clase - relación. El diagrama de agente es traducido a un diagrama de clase - relación (CDR), agregando los lazos.
- Los diagramas físicos de interacción, estos son creados con cada proceso principal, basado en los diagramas lógicos.

### 4.1. Diagramas Internos

En este diagrama se hace referencia solamente a los agentes internos.



### 4.2. Diagrama de Clase - Relación

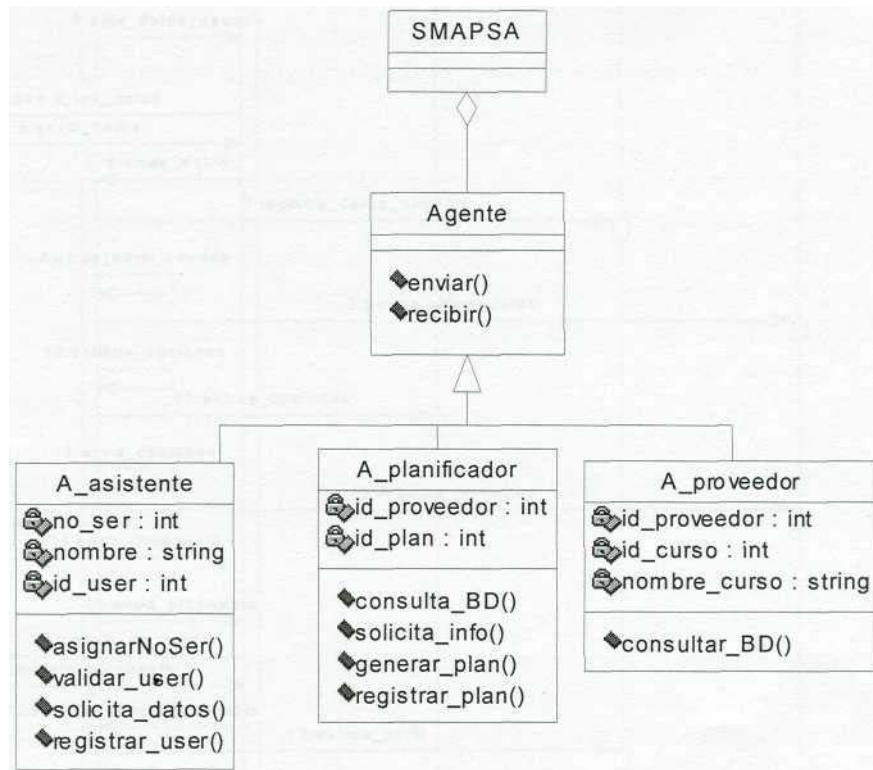
La primera cosa a acentuar es que los agentes, como los objetos, son casos de clases; así un diagrama de clase - relación, usado en metodologías Orientado a Objetos, es también un componente esencial de nuestra metodología Orientado a Agentes.

El Sistema Multiagente para la Planificación y Supervisión del Aprendizaje (SMPSA), esta constituido por la clase Agente, esta clase comparte dos métodos los cuales son *enviar* y *recibir*. Las funciones principales de *enviar* es la transmitir datos hacia las demás entidades y la de *recibir* permite que las clases puedan recibir datos de otras clases.

La clase *A\_asistente* permite asignar un numero de servicio, validar a los usuarios, además de solicitar y guardar en la BD los datos del usuario.

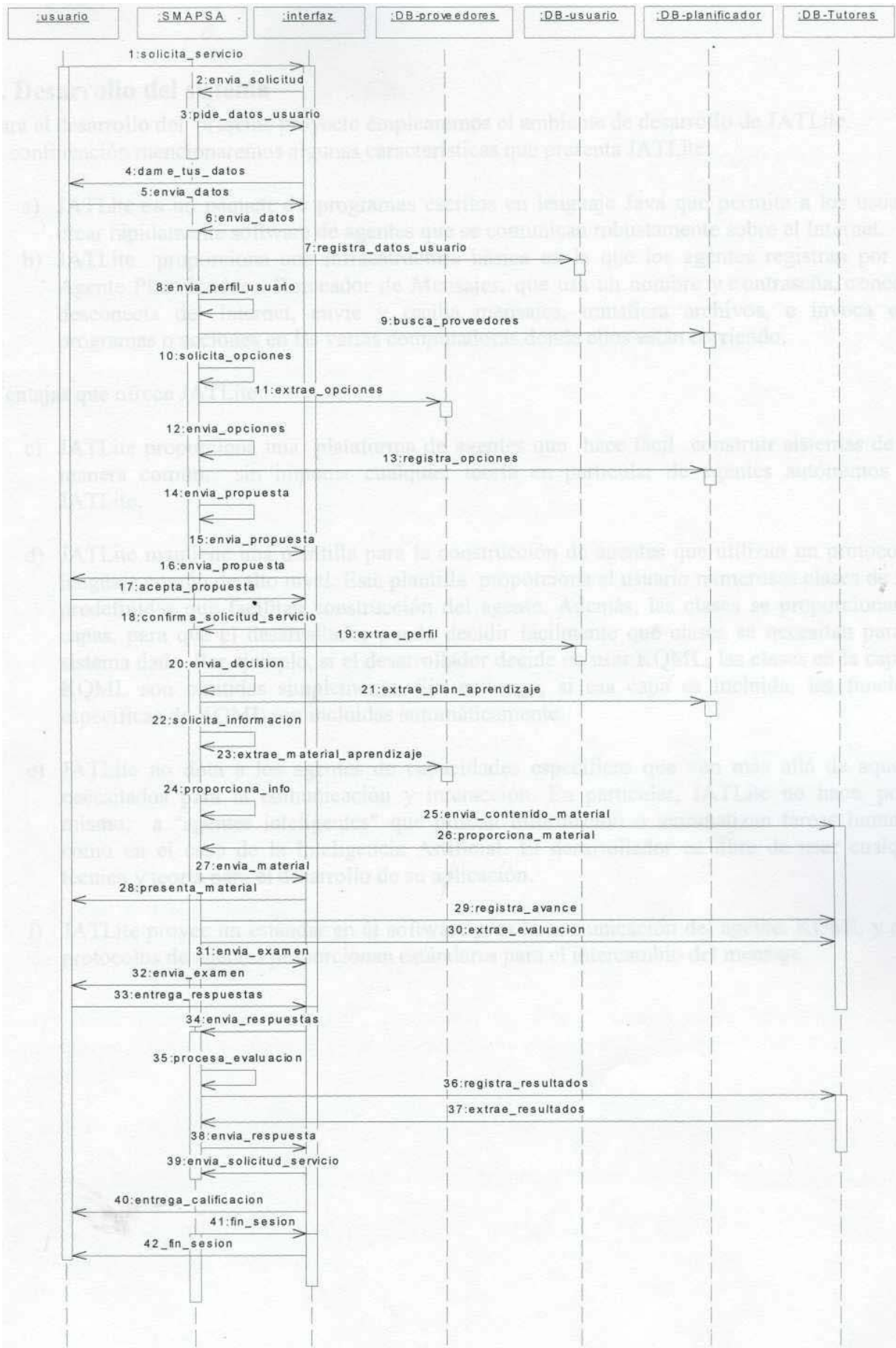
La clase *A\_planificador* sus funciones principales son: consultar en la BD los proveedores con que cuenta, solicitar información a los proveedores, generar y registrar el plan de aprendizaje.

La clase *A\_proveedor* permite consultar en la bases de datos, enviando la información solicitada por la clase *A\_planificador*.



### 4.3. Diagramas de interacción física

En esta etapa los diagramas lógicos de interacción se refinan, y se desarrollan diagramas físicos de interacción. Los diagramas de interacción se traducen a una serie de métodos invocados (es decir, una serie de funciones miembro llamadas). Para detallar la interacción que existe en el dominio, se muestra la interacción de los agentes participantes y a continuación se detalla.



## 5. Desarrollo del sistema

Para el desarrollo del presente proyecto emplearemos el ambiente de desarrollo de JATLite.

A continuación mencionaremos algunas características que presenta JATLite:

- a) JATLite es un paquete de programas escritos en lenguaje Java que permite a los usuarios crear rápidamente software de agentes que se comunican robustamente sobre el Internet.
- b) JATLite proporciona una infraestructura básica en la que los agentes registran por un Agente Planificador - Routeador de Mensajes, que usa un nombre y contraseña, conecta / desconecta del Internet, envía y recibe mensajes, transfiere archivos, e invoca otros programas o acciones en las varias computadoras donde ellos están corriendo.

Ventajas que ofrece JATLite:

- c) JATLite proporciona una plataforma de agentes que hace fácil construir sistemas de una manera común, sin imponer cualquier teoría en particular de agentes autónomos en JATLite.
- d) JATLite mantiene una plantilla para la construcción de agentes que utilizan un protocolo y lenguaje común de alto nivel. Esta plantilla proporciona al usuario numerosas clases de Java predefinidas que facilitan construcción del agente. Además, las clases se proporcionan en capas, para que el desarrollador pueda decidir fácilmente qué clases se necesitan para un sistema dado. Por ejemplo, si el desarrollador decide no usar KQML, las clases en la capa de KQML son omitidas simplemente. Sin embargo, si esa capa es incluida, las funciones específicas de KQML son incluidas automáticamente.
- e) JATLite no dota a los agentes de capacidades específicas que van más allá de aquellos necesitados para la comunicación y interacción. En particular, JATLite no hace, por sí mismo, a "agentes inteligentes" que buscan información o automatizan tareas humanas, como en el caso de la Inteligencia Artificial. El desarrollador es libre de usar cualquier técnica y teoría para el desarrollo de su aplicación.
- f) JATLite provee un estándar en el software para la comunicación del agente. KQML y otros protocolos de agentes proporcionan estándares para el intercambio del mensaje.

## 6. Bibliografía

[Agent Builderl, 2002]Agent Buider,(2002) "An integrate Toolkit for Constructing Intelligent Software Agents". Guía de usuario , version 1.3 , 30 de abril del 2002. Reticular System, Inc. <http://www.agentbuilder.com>

"La Revolución Informática y la Educación ", <http://msip.lce.org/erporto/libros/edul/capi2.htm>

[Murray, 98] Murray L.(1998) "Nuevas Tecnologías en la enseñanza - aprendizaje" Hemeroteca Virtual ANUIES, México, D.F., 1998. <http://www.hemerodigital.unam.mx/ANUIES>

[Ayala,2000] Ayala, Gerardo. (2000) "Ambiente de Aprendizaje Basados en Agentes Inteligentes." Memorias Congreso Virtual Computo 2000, México D. F., 2000 <http://www.computo2000.unam.mx/inteligencia/ayala/v00.htm>

[Hietala et al, 99] Hietala, P. Y Niemirepo, T. (1999) "Studying learner - computer interaction in agent bases social learning environments", University of Tampere, Tampere Finland, 1999

MADKIT, (a Multi Agent Development, kit). <http://www.madkit.org>

[Núñez et al, 99] Núñez, G., Sheremetov, L., Guzmán, A. (1999) "Tecnologías de Inteligencia Artificial y de Agentes Computacionales en la Educación: El Proyecto EVA". CIC - IPN, México D.F., 1999 <http://eva.cic.ipn.mx/~public/online/academia/Academia2.html>

[Sheremetov & Núñez, 99] Sheremetov, L. y Núñez, G. (1999) "Multi - Agent Framework for Virtual Learning Spaces" Selected Papers of Research 1999, Published in the International Journal of Interactive Learning Research (JILR), N 3-4

[Aroyo & Kommers, 99] Aroyo,L. y Kommers, P. (1999) "Los Agentes Inteligentes para la Computación Educativa" Periódico de Investigación de Aprendizaje Interactivo. Volumen 10, Número <sup>3</sup>/<sub>4</sub>1999.

JADE (Java Agent Development Framework). <http://sharon.cselt.it/proiects/jade/>

[1] JATLite Beta Complete Documentation (1998), Stanford University, <http://java.stanford.edu>

[Heecheol et al, 2000] Heecheol, J. Petrie, C. y Cutkosky M. R. "JATLite: A Java Agent Infrastructure with Message Routing".

[Nwana & Ndumu, 98] Nwana, H. Y Ndumu, D. (1998) "A Brief Introduction to Software Agent Technology".

Tomado de:

[Jeannings & Wooldridge, 98 ] Jeannings, N. Y Wooldridge M.(1998) "Agent Technology Foundations, Applications, and Markets". Ed. Springer 1998.

[Bigus & Bigus,2001] Joseph P. Bigus, Jennifer Bigus. (2001),"Contracting Intelligent Agents Using Java", Second Edition, John Wiley & Sons Inc. 2001.

[O'Brien & Nicol, 1998] PD O'Brien , R C Nicol. (1998),"Towards a Standar for Software Agents", Tomado de la página de FIPA.

[Wooldridge & Jeannings, 94 ] Wooldridge M. y Jeannings, N. (1994) "Intelligent Agents: Theory and Practice". Registrado en Knowledge Engineering Review , Octubre 1994, Revisado en 1995.

[2] Foundation for Intelligent Physical Agents (FIPA), <http://www.fipa.org/about/index.html>

[Blum & Furst, 1997] Blum, L. y Furst, L. (1997) "Fast Planning Through Planning Graph Analysis" Tomado de la Artificial Intelligence School Of Computer Science Cargenie Mellon University Pittsburgh.

[Hannebauer & Geske, 1999] Hannebauer , M. y Geske, Ulrich. (1999) "Coordination Distributed CLP-Solvers in Medical Appointment Scheduling", Planning and Optimization Laboratory.

[Dean & Kambhanpati, 1996] Dean , T. y Kambhampati, S. (1996) "Planning and Scheduling" Tomado de la CRC Handbook of Computer Science and Engineering.

[Dorn & Kerr, 199 ] Dorn , J. y Kerr, R. (199) "Co-operating Scheduling Systems Communicating Through Fuzzy Sets".