

MODELADO E IMPLEMENTACIÓN DE UN SISTEMA MULTIAGENTE PARA EL DIAGNÓSTICO DE ENFERMEDADES DE TRANSMISIÓN SEXUAL¹

MODELLING AND IMPLEMENTATION OF A MULTI-AGENT SYSTEM FOR DIAGNOSING SEXUALLY TRANSMITTED DISEASES

ADAPTAÇÃO E IMPLEMENTAÇÃO DE UM SISTEMA MULTIAGENTE PARA O DIAGNÓSTICO DE DOENÇAS SEXUALMENTE TRANSMISSÍVEIS

Mauro Callejas Cuervo

Ingeniero de sistemas, especialista en ingeniería de software, magister en ciencias computacionales y actualmente desarrollando tesis de Doctorado en Ciencia y Tecnología Informática en la Universidad Carlos III de Madrid España. Director del Grupo de Investigación en Software, GIS-UPTC. Profesor Asistente, Universidad Pedagógica y Tecnológica de Colombia - UPTC, Facultad de Ingeniería, Escuela de Sistemas y Computación. Colombia maurocallejas@yahoo.com, mauro.callejas@uptc.edu.co

Liliana Milena Parada Prieto

Ingeniera de sistemas y computación. Asesora del Instituto Colombiano de Bienestar Familiar. Integrante del Grupo de Investigación en Software, GIS Escuela de Ingeniería de Sistemas y Computación, Universidad Pedagógica y Tecnológica de Colombia - UPTC, Colombia lilianamparada@gmail.com

Andrea Catherine Alarcón Aldana

Ingeniera de sistemas y computación. Especialista en Ingeniería de Software. Magister en Software Libre. Integrante del Grupo de Investigación en Software - GIS. Profesor asistente, Facultad de Ingeniería, Universidad Pedagógica y Tecnológica de Colombia - UPTC, Colombia. Escuela de Ingeniería de Sistemas y Computación, Universidad Pedagógica y Tecnológica de Colombia - UPTC, Colombia acalarcon@gmail.com

Resumen

Esta investigación presenta el modelado e implementación de una aplicación informática basada en un sistema multiagente como herramienta de apoyo a programas de prevención de enfermedades de transmisión sexual (ETS). Para ello se abordó el proceso de investigación concerniente al área de los agentes de software y los sistemas multiagentes, así como también las características que identifican las enfermedades de transmisión sexual; finalmente se exponen las principales fases del desarrollo de una aplicación web basado en sistemas multiagentes para la prevención de estas enfermedades, siguiendo la metodología Ingenias y usando la plataforma Java Agent Development Framework.

Abstract

This research work discusses modeling and implementation of a software application based on a multi-agent system that can be used as a supporting instrument in STD (sexually transmitted disease) prevention programs. To this end, we addressed the research process relative to the field of software agents and multi-agent systems, including a review of characteristics that describe sexually transmitted diseases. It also provides a discussion of the primary phases of development of a web application based on multi-agent systems for the prevention of these kinds of diseases, following the INGENIAS approach and using the Java agent development framework platform.

Palabras clave

Intelligent agents, sexually transmitted diseases, agents' platforms, multi-agent systems.

Keywords

Intelligent agents, sexually transmitted diseases, agents' platforms, multi-agent systems.

Fecha de recepción: 06 - 02- 2012

Fecha de aceptación: 08 - 05 - 2012

Resumo

Esta investigação apresenta a adaptação e implementação de um aplicativo informático baseado em um sistema multiagente como ferramenta de apoio a programas de prevenção de doenças sexualmente transmissíveis (DST). Para isso foi abordado o processo de investigação com relação à área dos agentes de software e dos sistemas multiagentes, bem como as características que identificam as doenças sexualmente transmissíveis;

finalmente, são expostas as principais fases de desenvolvimento de um aplicativo Web baseado em sistemas multiagentes para a prevenção dessas doenças, seguindo a metodologia Ingenias e usando a plataforma Java Agent Development Framework.

Palavras-chave

Agentes Inteligentes, doenças sexualmente transmissíveis, plataformas de agentes, sistemas multiagentes.

Introducción

El desarrollo de software ha buscado siempre sistematizar y optimizar procesos llevados a cabo por el ser humano, según mencionan Maturana, Ferrer y Baranao (2004), es por esta razón que esta investigación se realizó con el ánimo de lograr un acercamiento al paradigma que intenta simular el comportamiento humano, así como en otras áreas lo han tratado autores como Moreno, Oña y Martínez (1998) y Romeo y Pastor (2004), más allá de desarrollar un sistema complejo de agentes, como lo planteado por Sansores y Pavón (2005), es así que en este artículo se enfatiza en la importancia de esta área, mediante una indagación tanto en el estado del arte como en las teorías involucradas.

Aunque a nivel colombiano ya se está abarcando la investigación en Agentes y Sistemas Multi-Agentes (SMA), en las universidades del departamento de Boyacá, Colombia, aún no se ha profundizado en esta área de investigación, así como también es de mencionar que de acuerdo con estudios realizados por la Unidad de Promoción de la Salud y Prevención de la Enfermedad, de la Universidad Pedagógica y Tecnológica de Colombia (UPTC), las Enfermedades de Transmisión Sexual (ETS), son las más frecuentes y alarmantes en la comunidad universitaria y actualmente no existe una herramienta que brinde soporte tecnológico a programas enfocados en la prevención de dichas enfermedades.

La implantación de una herramienta de apoyo para la prevención de Enfermedades de Transmisión Sexual, permite a la Unidad de Promoción y Prevención de la UPTC un desempeño óptimo en el desarrollo de sus objetivos y actividades; además de lograr un mayor cubrimiento en la población universitaria y contribuir

a evitar complicaciones innecesarias por falta de información o conocimiento sobre diferentes problemas de salud.

Este documento presenta inicialmente los trabajos relacionados con sistemas multiagentes en el campo de la salud, posteriormente se expone la fundamentación teórica para la investigación, a continuación se describen los resultados obtenidos en desarrollo de la aplicación de Sistemas MultiAgentes (SMA) en una herramienta Web de apoyo a programas de prevención de enfermedades de transmisión sexual en la UPTC y finalmente se plasman las conclusiones.

1. Trabajos relacionados

En la actualidad, la temática que involucra agentes de software, constituye una de las más importantes áreas de investigación. Asimismo los adelantos en las aplicaciones de la inteligencia artificial distribuida provocan un creciente interés en la exploración de los sistemas multiagente, así como de su aplicación en la resolución cooperativa de problemas.

Existe un gran número de grupos de investigación, productos implantados, artículos, congresos y conferencias que se han realizado respecto al paradigma de los agentes y los SMA, razón por la cual no debe ser un tema aislado del área de investigación de la comunidad estudiantil. A continuación se presentan algunos de los logros obtenidos en el campo de investigación de agentes inteligentes en el área de la salud.

En Corchado y otros (2008) se presenta el desarrollo de un agente autónomo inteligente para el seguimiento de la atención de la salud de pacientes de Alzheimer en

tiempo real en residencias geriátricas. El agente opera en dispositivos inalámbricos y es una buena opción para la organización del trabajo de las enfermeras que tienen la misión de cuidar de estos pacientes.

En Crutzen y otros (2011), se revela la implementación de un chatbot, sistema basado en inteligencia artificial, específicamente un agente. El uso de un chatbot, en el campo de la promoción de la salud tiene una gran importancia debido a que puede llegar a un grupo significativo de adolescentes y darles respuestas a preguntas relacionadas con sexo, drogas y alcohol, temas que en otros escenarios no son tratados con la facilidad que ofrece el sistema.

En el documento presentado por Edwards y otros (2011), se propone un enfoque basado en agentes inteligentes para reproducir las operaciones de búsqueda de información acerca de hospitales y centros de salud existentes en una localidad específica con el propósito de solicitar una cita médica, de la misma manera que lo haría un individuo en condiciones de rutina (de forma manual). El sistema propuesto fue simulado y validado a través de la aplicación en el teléfono inteligente de un individuo usando el kit de desarrollo con agentes JADE-LEAP.

De la misma forma Jara y otros (2011), tratan lo relacionado con la evaluación de tres métodos de aprendizaje automático (con técnicas de inteligencia artificial) para clasificar diagnósticos de neoplasias, comentan que los diagnósticos médicos son una fuente valiosa de información para evaluar el funcionamiento de un sistema de salud y afirman que su uso en sistemas de información es difícil porque estos se encuentran normalmente escritos en lenguaje natural.

En estudios más recientes (Lasheng *et al.*, 2012) se propone un sistema multiagente basado en la integración de procesos de vigilancia de la salud (IHMS), que está compuesto por redes inalámbricas de sensores, redes públicas de comunicación, múltiples agentes y tecnologías de servicios web para uso de adultos mayores desde el hogar. El sistema ha sido diseñado con base en cinco plataformas de servicios: el hogar, servicios comunitarios, servicio médico de salud, personal a cargo del cuidado del adulto mayor y centro de datos. Todas las plataformas de servicios se implementan a través de roles distintos, agentes que pueden ser deliberativos o reactivos. El propósito del sistema es proveer a las

personas mayores una atención de salud móvil, ubicua y personalizada, así como mejorar su calidad de vida a través de la implementación de tecnología basada en agentes inteligentes.

2. Fundamentación teórica

Para el desarrollo de la investigación fue necesario contar con la comprensión de los soportes y fundamentos teóricos, por lo cual, a continuación se presenta la contextualización de la investigación.

2.1. INTELIGENCIA ARTIFICIAL DISTRIBUIDA

La Inteligencia Artificial Distribuida (IAD) es la rama de la Inteligencia Artificial (IA) que trata la resolución de problemas de manera cooperativa en un cierto ambiente, por medio de entidades, que según Shoham (1993), son denominadas agentes.

2.2. AGENTES Y TAXONOMÍA

Existen múltiples definiciones de agente, sin embargo, aún no se ha concertado un concepto consolidado. Una de las más acertadas es la que presentan Wooldridge y Jennings (1995), que dice: "Es un sistema de cómputo, situado en un ambiente cualquiera, y que es capaz de realizar acciones autónomas que afectan su ambiente de acuerdo con ciertos objetivos de diseño". En cuanto a la taxonomía de los agentes, no existe un consenso sobre su definición, sin embargo es de mencionar que el concepto de "agente" tiene asociado un conjunto de cualidades, las más relevantes son: autonomía, reactividad, proactividad, habilidad social, entre otras, según lo mencionado por Wooldridge (2009).

2.3. SISTEMAS MULTIAGENTES (SMA)

Así como sucede con la definición de "agente", tampoco existe un concepto único para Sistema MultiAgente. Wooldridge (2009), expresa que los SMA, son sistemas compuestos por múltiples elementos computacionales que interactúan entre sí, denominados agentes, que se encargan de la coordinación de la conducta inteligente de un grupo de agentes autónomos, los cuales tienen la capacidad de coordinar su conocimiento, objetivos, habilidades, toma de decisión y planes (Unal, 2012).

2.4. ARQUITECTURAS DE AGENTES

Según la ingeniería del software, una arquitectura se encarga de la división del sistema en módulos y la descripción de cada uno de ellos, especificando además la interrelación que hay entre estos. Para el caso de los sistemas basados en agentes existen arquitecturas como:

2.4.1. DELIBERATIVAS

En *Agent Theories, Architectures, and Languages: A Survey*, los autores Wooldridge y Jennings (2012), presentan una arquitectura de agente deliberativo, que contiene un mundo representado explícitamente y un modelo lógico del mismo, y en la cual las decisiones (por ejemplo, acerca de las acciones a realizar) son tomadas por medio de un razonamiento lógico (o por lo menos pseudo-lógico), basado en concordancia de patrones y manipulación simbólica.

2.4.2. REACTIVAS

Las arquitecturas reactivas no tienen un modelo del mundo simbólico como elemento central de razonamiento y no utilizan razonamiento simbólico complejo, sino que siguen un procesamiento ascendente (bottom up), para lo cual mantienen una serie de patrones que se activan bajo ciertas condiciones de los sensores y tienen un efecto directo en los actuadores (Universidad Politécnica de Madrid, 2012).

2.4.3. HÍBRIDAS

Estas arquitecturas presentan varios subsistemas, en los que se combinan las dos anteriores; unos deliberativos para resolver tareas que requieren un modelo simbólico y otros reactivos para responder ante estímulos que no precisan deliberación. (Universidad Politécnica de Madrid, 2012).

2.5. ONTOLOGÍAS

Una definición concreta para este término la presenta Jiménez (2011), mencionando que una ontología “es una especificación de objetos, conceptos y relaciones de una determinada área de interés”. La utilización

de ontologías busca favorecer la comunicación entre personas, organizaciones y aplicaciones, y de esta forma lograr la interoperabilidad entre sistemas informáticos y razonar automáticamente. Las ontologías aplicadas a los SMA permiten que todos los agentes que compartan una misma ontología para la representación de conocimiento, tengan un entendimiento de palabras en el lenguaje de comunicación que usen.

2.6. LENGUAJE DE COMUNICACIÓN FIPA-ACL.

La comunicación entre agentes es la base para las interacciones y la organización de un SMA, de modo que según Molina, García y Bernardos (2011), Agent Communication Language (ACL) resultó de la necesidad de tener un lenguaje que permitiera la interacción entre agentes autónomos distribuidos. ACL tiene tres componentes: un vocabulario, un lenguaje de contenido llamado KIF (Knowledge Interchange Format) y un lenguaje de comunicación llamado KQML (Knowledge Query Manipulation Language). Un mensaje de ACL, de acuerdo con lo mencionado por Cortés (2011), es un mensaje en KQML que se compone de una directiva de comunicación y un contenido semántico en KIF expresado en términos del vocabulario.

2.7. PLATAFORMAS ORIENTADAS A AGENTES

Existe variedad de opciones para la selección de una metodología, así mismo las hay para la elección de la plataforma; con el paso de los años se han desarrollado varias plataformas, cada una de ellas con ciertas características específicas, por lo cual a continuación se presenta una descripción de las más representativas, según Marchetti y García (2003):

- JACK: Entre las características relevantes, se puede mencionar que esta plataforma utiliza una arquitectura base denominada BDI (Beliefs-Desire-Intention, tipo de agente racional cuyo comportamiento es dirigido por creencias, deseos e intenciones), soporta cualquier tipo de agentes, para implementación de agentes soporta el lenguaje también denominado Jack, además la documentación disponible es muy completa.

- **MADKit:** El ACL ó Lenguaje de Comunicación entre agentes que soporta es KQML, y al igual que la plataforma Jack, soporta cualquier tipo de agentes. Una de sus ventajas es que soporta varios lenguajes para implementación de agentes, entre los cuales se puede mencionar java, jess, Scheme, BeanShell y python, principalmente. También es de mencionar que esta plataforma es liberada con licencia GPL/LGPL, pero presenta aún poca documentación.
- **ZEUS:** Esta plataforma también soporta KQML como ACL, y al igual que Jack, su arquitectura base es BDI; solamente soporta agentes deliberativos y colaborativos, y el lenguaje para la implementación de agentes es java, se distribuye bajo licencia Mozilla public, pero la documentación disponible es pobre.
- **JADE (Java Agent Development Framework):** La arquitectura de comunicación de esta plataforma crea y maneja una cola de mensajes ACL entrantes, debido a que este entorno simplifica la implementación de SMA mediante una capa de soporte (middle-ware). Además es de mencionar que soporta cualquier tipo de agentes, los cuales son implementados utilizando el lenguaje java; esta plataforma es liberada bajo licencia LGPL y ofrece una completa documentación. Jade (2012).
- **AgentBuilder:** Esta plataforma, al igual que otras descritas, presenta arquitectura base BDI: es posible utilizarla bajo licencia de versión de evaluación libre limitada y revela una documentación completa.

En la Tabla 1 se presenta una comparación simple de las plataformas más comunes de desarrollo de software basado en agentes, teniendo como referente lo planteado por Marchetti y García (2003).

2.8. METODOLOGÍAS DE DESARROLLO DE SISTEMAS BASADOS EN AGENTES

En el proceso de desarrollo de software basado en agentes es necesario tener en cuenta una metodología que guíe esta actividad; actualmente existen múltiples opciones entre las cuales se destacan: MASE: Multi-agentsystems Software Engineering (Scott et al., 2001), INGENIAS (Gómez-Sanz et al., 2002), TROPOS (Bresciani et al., 2004), BDI (Bratman, 1987; Kinny et al. 1995), GAIA (Wooldridge et al., 2000), MAS-CommonKADS (Iglesias, 1998).

3. Resultados

Teniendo como base la información aportada en el numeral anterior, se definió como metodología de desarrollo INGENIAS, como arquitectura seleccionada deliberativa y la plataforma usada fue Java Agent Development Framework.

A continuación se describen, los modelos requeridos para el desarrollo del sistema multiagente.

Criterios / Plataforma	JACK	MADKit	ZEUS	JADE	AgentBuilder
ACL soportado	-----	KQML	KQML	ACL.	-----
Arquitectura base	BDI	Agente/grupo/rol	BDI	-----	BDI
Tipo de agentes soportados	Cualquiera	Cualquiera	Deliberativos colaborativos	Cualquiera	Agentes inteligentes
Lenguajes soportados para implementación de agentes.	Jack	Java, Jess, Python, Scheme, BeanShell	Java	Java. Java(J2ME)	-----
Movilidad de código	No detalla	No detalla	No detalla	Migración débil	No detalla
Disponibilidad	on-line	on-line	on-line	on-line	-----
Licencia	gratis 30 dias	GPL/LGPL	Mozilla public	LGPL.	Versión de evaluación libre, limitada
Interface	GUI	GUI	GUI	GUI	GUI
Instalación	Simple	Simple	Simple		
Documentación	Muy complete	Pobre	Pobre	Complete	Muy completa
Ayuda	Manual	On-line	Manual	Manual	Manual

Tabla 1. Comparación de plataformas.

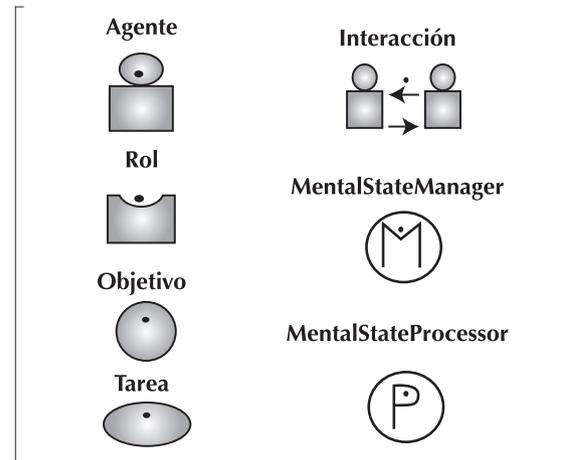
Fuente: Modificado de Marchetti y García (2003)

3.1. DEFINICIÓN DE AGENTES, SU COMPORTAMIENTO Y ONTOLOGÍAS

Los modelos requeridos son los siguientes:

- Casos de uso
- Modelo de agentes
- Modelo de objetivos y tareas
- Modelo de interacción
- Modelo de entorno

La notación para representar los modelos se observa en la Gráfica 1



Gráfica 1. Notación para representar los modelos

3.1.1 CASOS DE USO

Mediante estos diagramas se plasman los requisitos funcionales del sistema, que serán base para la identificación de agentes y su contexto de interacción; a continuación se presentan los diagramas y su respectiva especificación. (Ver Diagramas 1, 2, 3 y 4 y Tablas 3, 4 y 5).

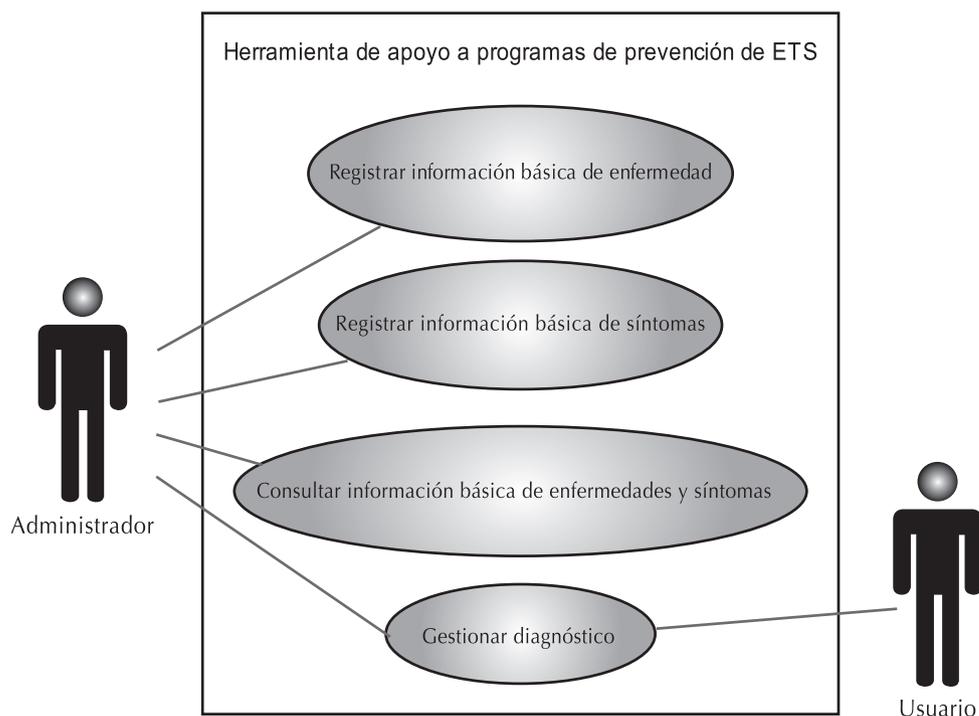


Diagrama 1. De contexto

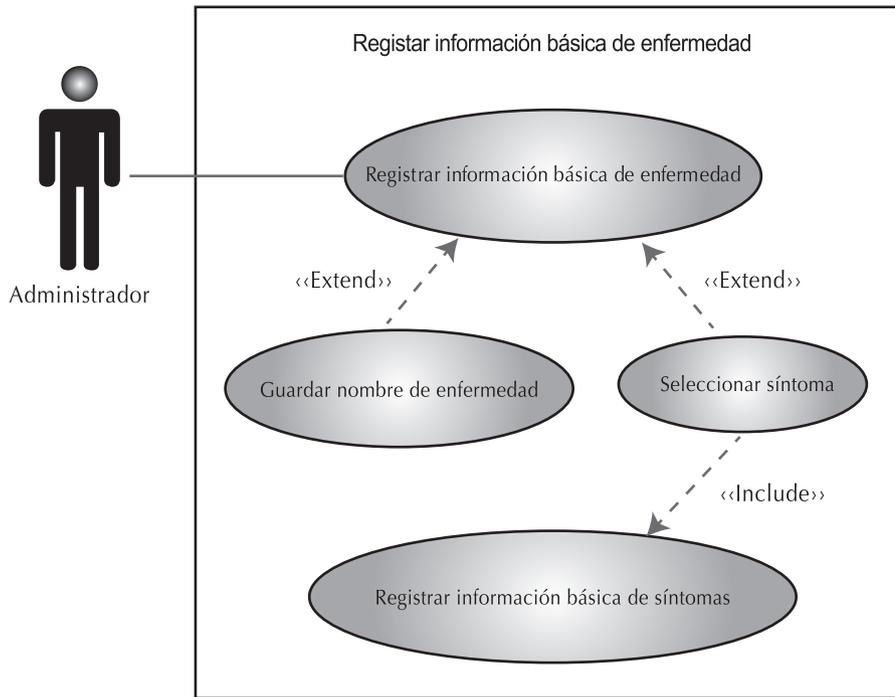


Diagrama 2. Caso de uso: registrar información de enfermedad

Nombre:	Registrar información básica de enfermedad/ CU-1	
Actor:	Administrador	
Descripción:	Describe el proceso para ingresar una nueva enfermedad al sistema.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Activa la función de ingresar nueva enfermedad	1. Muestra el formulario de nueva enfermedad
	2. Escribe el nombre de la nueva enfermedad y selecciona los síntomas de dicha enfermedad.	2. Acepta los datos y guarda la información en el sistema.
		3. Se reinicia el caso de uso
Alternativa:	1. Activa la función de ingresar nueva enfermedad	1. Muestra el formulario de nueva enfermedad
	2. Escribe el nombre de la nueva enfermedad y selecciona los síntomas de dicha enfermedad	2. Algún síntoma de la enfermedad no se ha ingresado al sistema
		3. Se reinicia el caso de uso
Precondición:	El administrador desea ingresar una nueva enfermedad al sistema.	
Poscondición:	La nueva enfermedad queda guardada y está lista para ser utilizada.	
Presunción:	La base de datos de ETS está disponible.	

Tabla 2. Registrar información de enfermedad

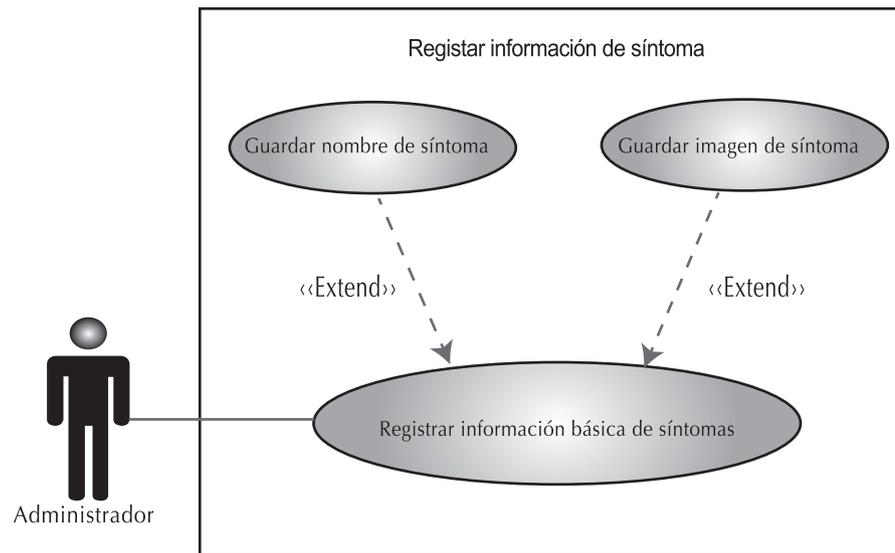


Diagrama 3. Caso de uso: registrar información de síntoma

Nombre:	Registrar información básica de síntoma/ CU-2	
Actor:	Administrador	
Descripción:	Describe el proceso para ingresar un nuevo síntoma al sistema.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Activa la función de ingresar nuevo síntoma	1. Muestra el formulario de nuevo síntoma
	2. Escribe el nombre del nuevo síntoma y selecciona una imagen relacionada con el síntoma.	2. Verifica la ubicación de la imagen y la carga al servidor.
		3. Guarda el nombre del síntoma y un enlace a la imagen.
		4. Se reinicia el caso de uso
Alternativa:	1. Activa la función de ingresar nuevo síntoma	1. Muestra el formulario de nuevo síntoma
	2. Escribe el nombre del nuevo síntoma y selecciona una imagen relacionada con el síntoma.	2. No logra subir la imagen al servidor.
		3. Se reinicia el caso de uso
Precondición:	El administrador desea ingresar un nuevo síntoma al sistema.	
Poscondición:	El nuevo síntoma queda guardado en el sistema	
Presunción:	La base de datos de ETS está disponible.	

Tabla 3. Registrar información de síntoma

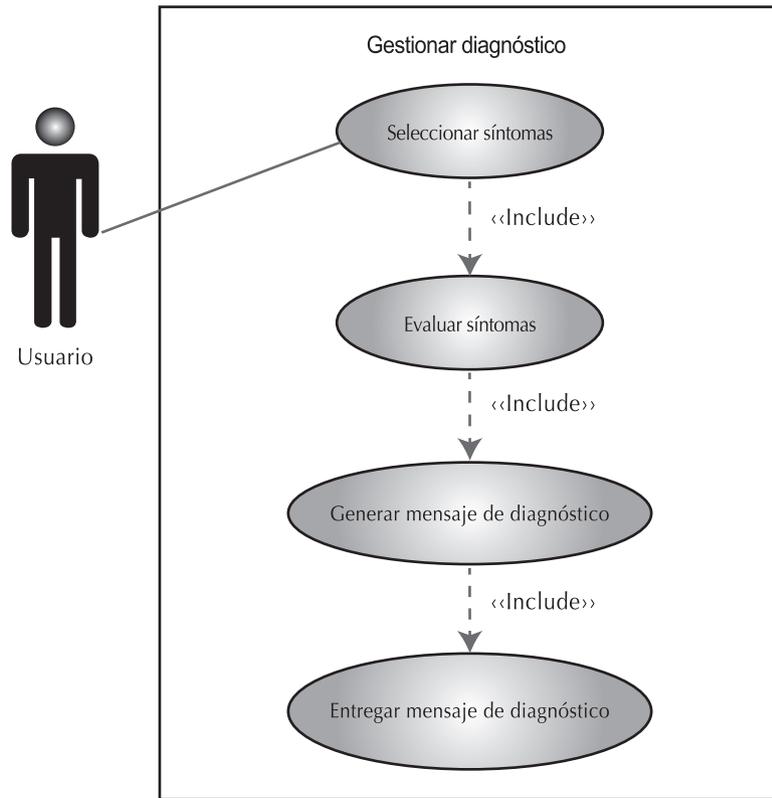


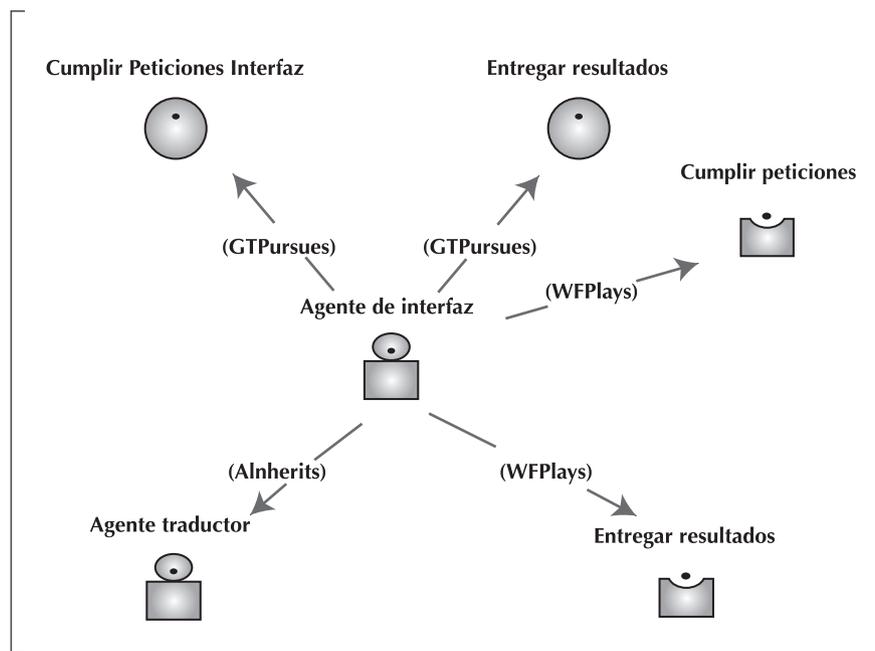
Diagrama 4. Caso de uso: gestionar diagnóstico

Nombre:	Gestionar diagnóstico/ CU-3	
Actor:	Usuario	
Descripción:	Describe el proceso para realizar un diagnóstico.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Activa la función realizar test de diagnóstico de ETS	1. Muestra los síntomas de forma grafica
	2. El usuario escoge los síntomas que padece	2. Analiza los síntomas recopilados.
		3. Genera un mensaje de las posibles enfermedades que padezca.
		4. Muestra el mensaje generado
Alternativa:	1. Activa la función realizar test de diagnóstico de ETS	1. No existe información de enfermedades en el sistema
		2. Se reinicia el caso de uso.
Precondición:	El usuario desea realizar el test de ETS.	
Poscondición:	Mensaje de posibilidad de infección de ETS	
Presunción:	La base de datos de ETS está disponible.	

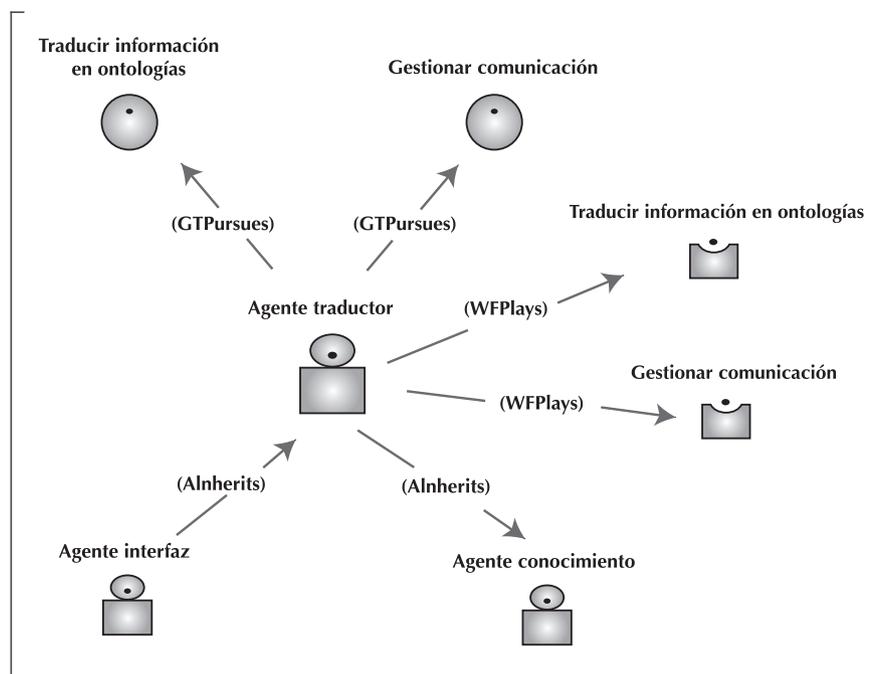
Tabla 4. Gestionar diagnóstico

3.1.2. MODELOS DE AGENTES

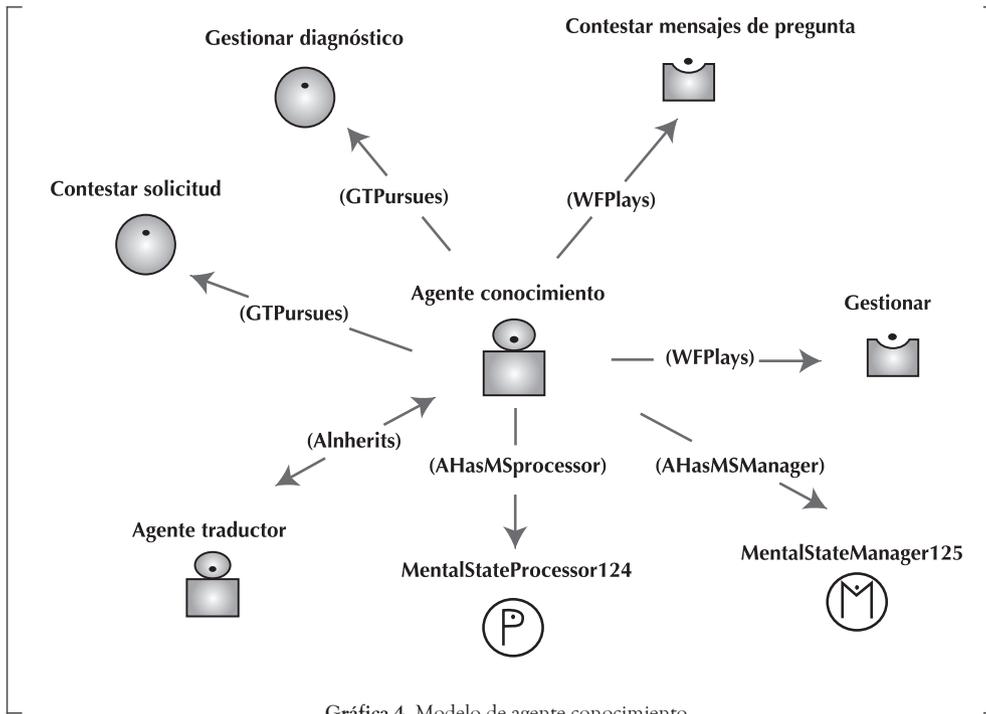
Mediante estos modelos se identifica y define qué agentes serán necesarios en la implementación del sistema propuesto. A continuación se presentan los modelos de los agentes: interfaz, traductor y conocimiento.



Gráfica 2. Modelo de agente conocimiento



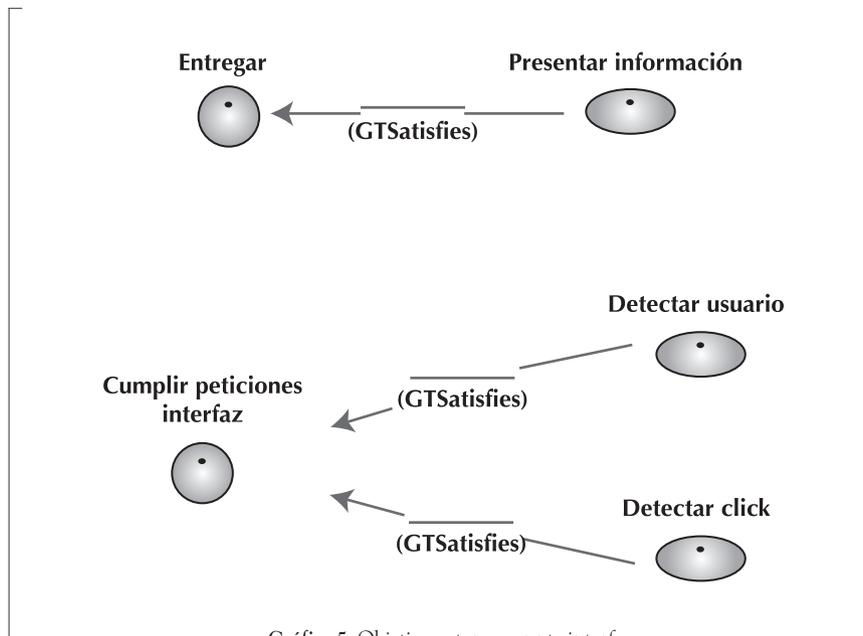
Gráfica 3. Modelo de agente traductor



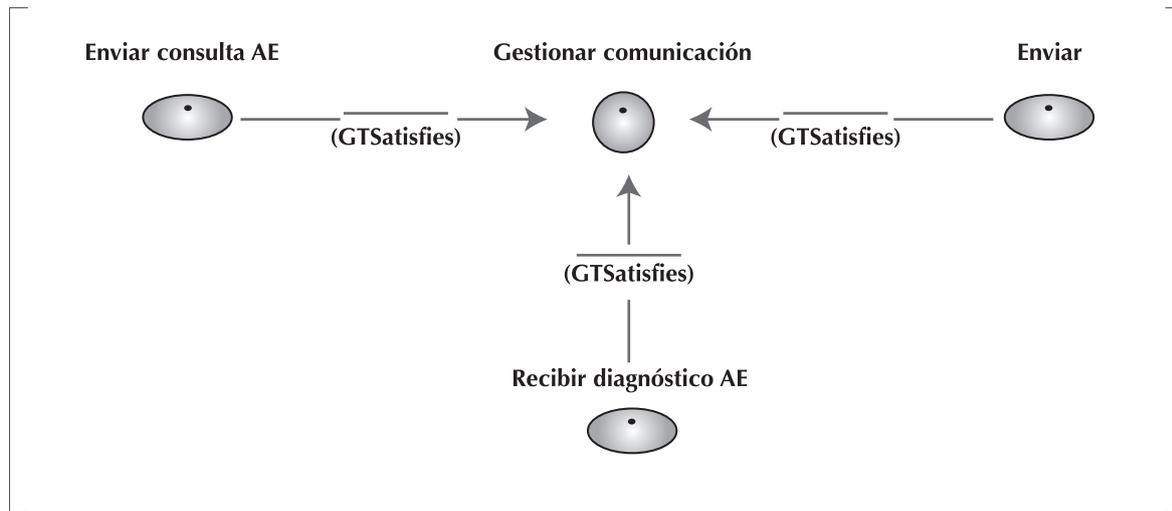
Gráfica 4. Modelo de agente conocimiento

3.1.3 MODELO DE OBJETIVOS Y TAREAS

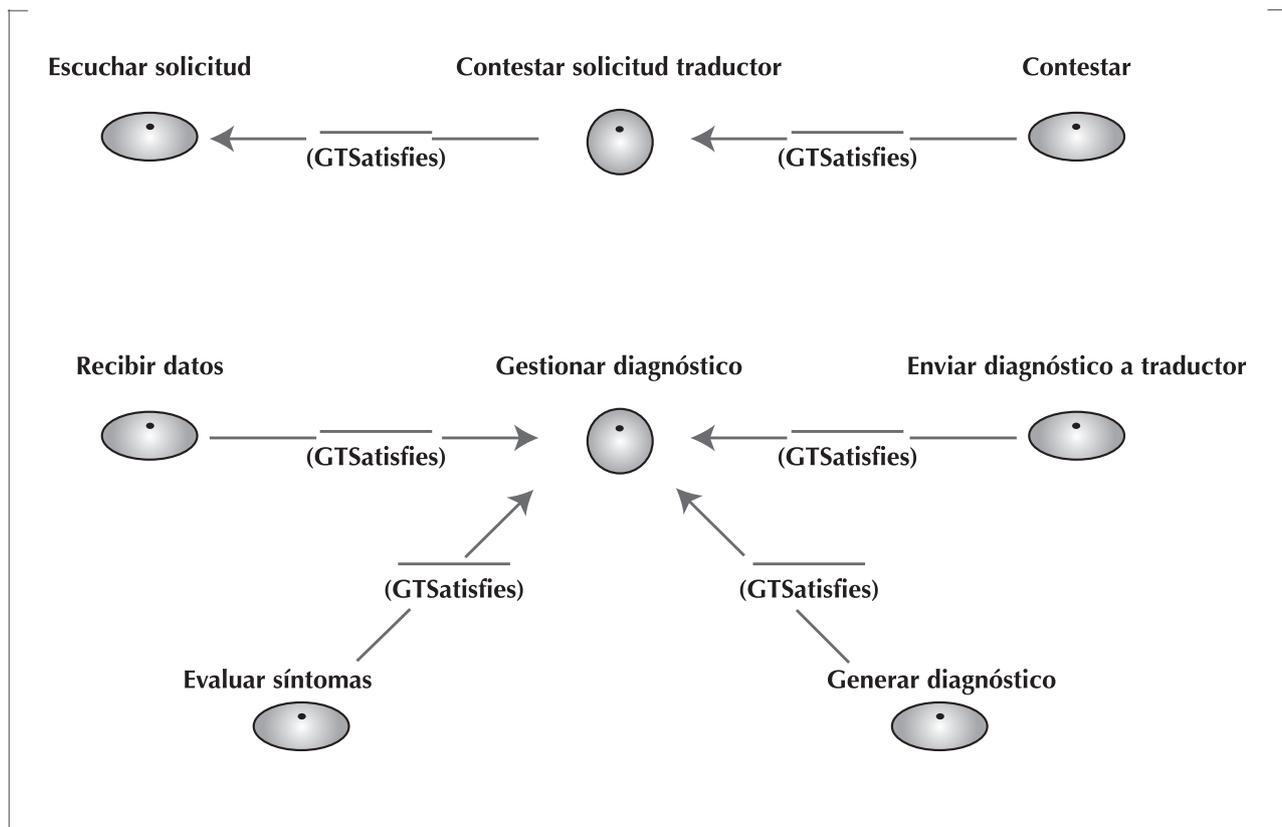
Cada uno de los agentes identificados anteriormente debe cumplir con una serie de objetivos y tareas para el cumplimiento funcional de la aplicación, los modelos de esta etapa se presentan a continuación:



Gráfica 5. Objetivos y tareas agente interfaz



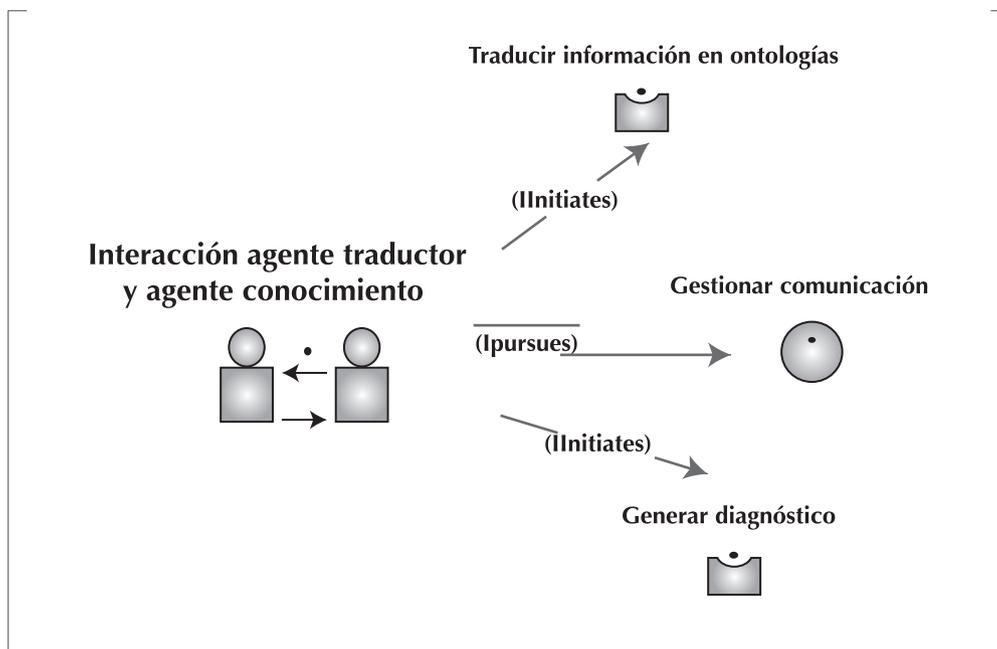
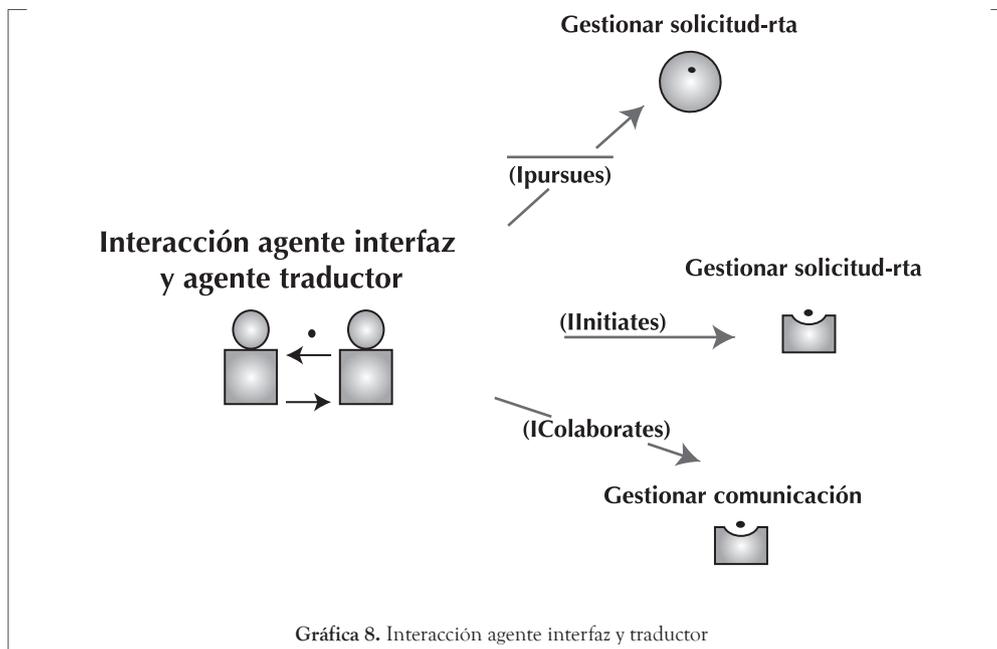
Gráfica 6. Objetivos y tareas agente traductor. Gestionar comunicación



Gráfica 7. Objetivos y tareas agente conocimiento

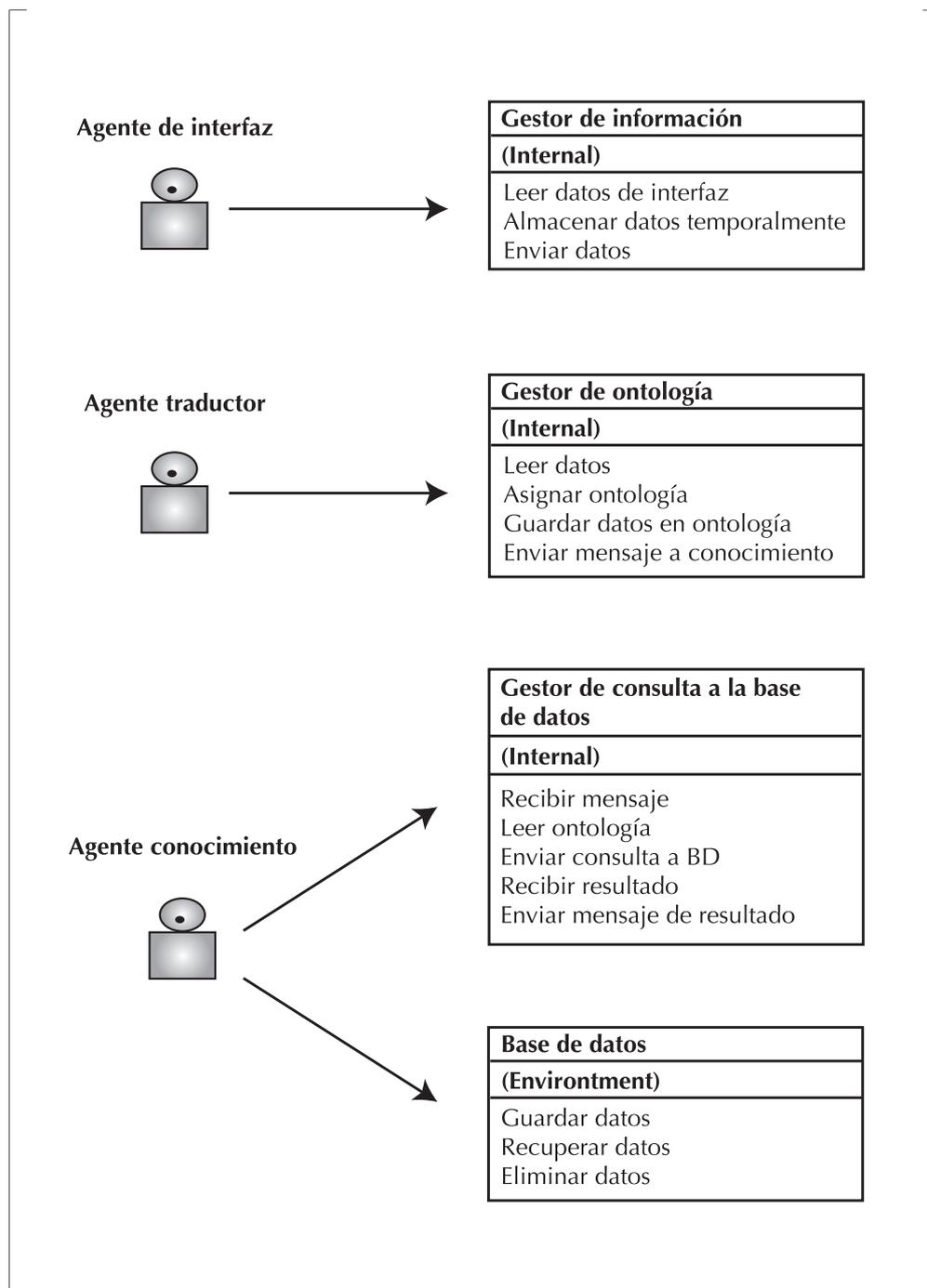
3.1.4 MODELOS DE INTERACCIÓN

En un sistema multiagente se tiene como parte fundamental la interacción entre los agentes identificados, este comportamiento se plasma con el uso de los modelos de interacción, los cuales son presentados a continuación:



3.1.5 MODELO DE ENTORNO

Haciendo uso de estos modelos es posible definir el tipo de recursos y aplicaciones a utilizar para el funcionamiento del sistema.



3.2. ONTOLOGÍA

Teniendo en cuenta la necesaria interacción entre los agentes, es fundamental realizar una especificación de una ontología que contenga los conceptos, predicado y acción a realizar por estos. El siguiente es un ejemplo de la implementación de una ontología en el sistema.

```

- Nombre de la Ontología: "ontología de síntoma"
- Vocabulario:
  public static final String SINTOMA = "síntoma";
  public static final String CODIGO = "CODIGO";
  public static final String NOMBRE = "NOMBRE";
  public static final String IMAGEN = "IMAGEN";
  public static final String SINTOMAS = "SINTOMAS";
  public static final String ENVIAR_A = "enviar_a";
  public static final String ENVIAR_A_ = "enviar_a_";
  public static final String ENVIAR = "enviar";
  public static final String ENVIAR_S = "enviar_s";
- Concepto: Síntoma
- Predicado: Enviar_a
- Acción: ENVIAR

```

3.3. DESCRIPCIÓN DE LOS MÓDULOS DEL SISTEMA WEB MULTIAGENTE

A continuación se presenta una descripción de los módulos que integran la aplicación multiagente.

3.3.1 MÓDULO ADMINISTRADOR

A través de este módulo el usuario administrador se encargará de gestionar la información de enfermedades y síntomas que estas presentan.(ver Figuras 1, 2, 3 y 4).

En la Figura 1 se visualiza la pantalla inicial del sistema, en donde se tiene la opción de ingresar a realizar la consulta o acceder al sistema como administrador.

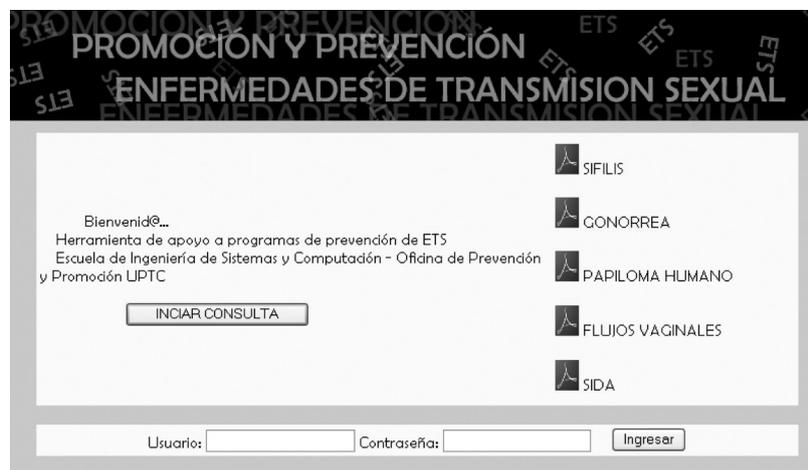


Figura 1. Pantalla inicial

En la Figura 2 se muestra la interface para guardar un nuevo síntoma, en donde se solicita un nombre y una imagen representativa de éste.

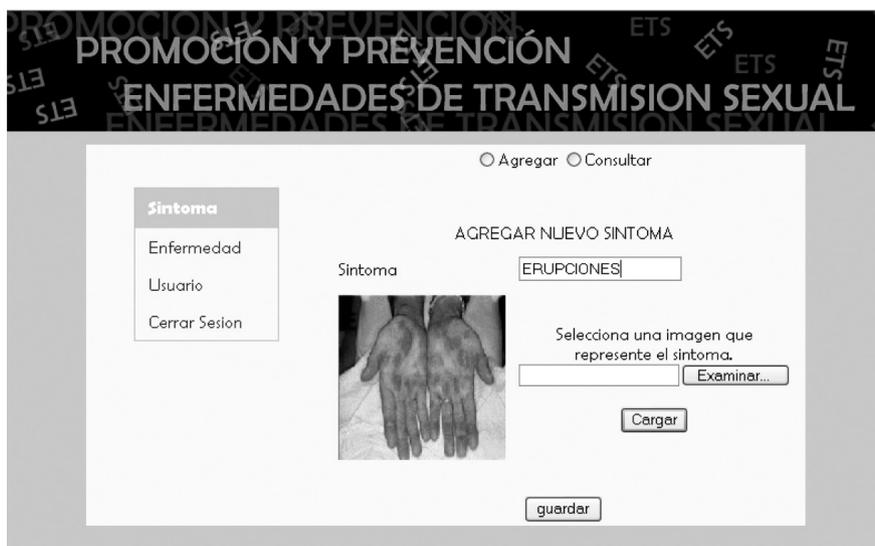


Figura 2. Pantalla guardar sintoma

Las Figuras 3 y 4, son las pantallas encargadas de ofrecer al administrador un listado de todos los síntomas y enfermedades respectivamente.



Figura 3. Pantalla consultar sintomas



Figura 4. Pantalla consultar enfermedades

3.3.2 MÓDULO CONSULTA

Mediante este módulo el usuario selecciona los síntomas que presenta o que requiere consultar y de acuerdo con dicha selección, los agentes se comunican y eliminan progresivamente los síntomas excluyentes, hasta poder dar un diagnóstico que sirva como referente para determinar qué enfermedad presenta, o a qué enfermedad se hace referencia e informa de algunas recomendaciones al respecto.

En la Figura 5 se visualizan los síntomas de forma gráfica, con el fin de que el usuario seleccione uno a uno los que presente; a medida que selecciona se eliminan los que sean excluyentes, (Figura 6).



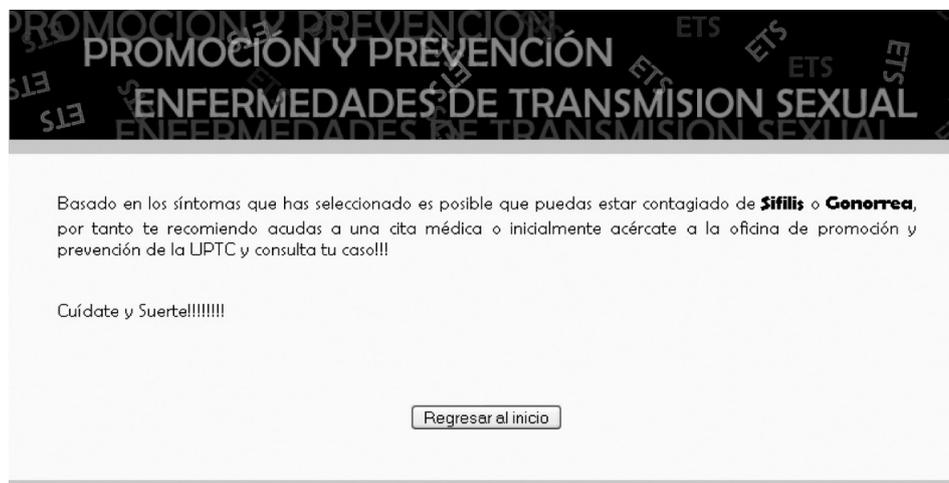
Figura 5. Pantalla selección de síntomas



Figura 6. Pantalla selección y eliminación de síntomas

Figura 7. Pantalla mensaje de resultado

Finalmente, en la Figura 7 se presenta una pantalla de resultado a la consulta, en el que se entrega al usuario un mensaje que contiene una(s) posible(s) ETS y la recomendación de acudir a un especialista.



4. Conclusiones

- Los resultados obtenidos en este proyecto ponen en evidencia que los agentes de software son actualmente un paradigma que cuenta con un nivel aceptable tanto de investigación como de desarrollo, es así que en el presente trabajo se demuestra la aplicabilidad de los SMA en el sector de la salud.
- Con el desarrollo de esta aplicación se logra obtener que la Unidad de Promoción y Prevención de la Universidad Pedagógica y Tecnológica de Colombia, tenga una herramienta tecnológica que pueda brindarle mayor cubrimiento sobre la población universitaria.
- Con el desarrollo de la presente investigación se logró analizar y diseñar un sistema Web, basado en sistema multiagentes para la prevención de ETS, para permitir que los estudiantes de la Escuela de Ingenierías de Sistemas y Computación abran un espacio más de investigación en el área de las ciencias computacionales.
- Por otra parte el sistema ofrece a la comunidad académica una herramienta que permite solucionar dudas y/o cuestionamientos sobre ETS, que no son fáciles de abordar frente a un especialista. ☰

REFERENCIAS BIBLIOGRÁFICAS

1. BRATMAN, M. Intentions, Plans, and Practical Reason. Harvard University Press. 1987.
2. BRESCIANI, P.; PERINI, A.; GIORGINI, P.; GIUNCHIGLIA, F. Tropos: An Agent-Oriented Software Development Methodology. In: Autonomous Agents and Multi-Agent Systems. Vol 8, N°. 3. (2004);pp. 203-236.
3. CORCHADO, J.; BAJO, J.; DE PAZ, Y.; TAPIA, D. Intelligent environment for monitoring Alzheimer patients, agent technology for health care, Decision Support Systems. Vol. 44. N°. 2. pp. 382-396. 2008. Disponible en: <http://www.sciencedirect.com/science/article/pii/S0167923607000760>. Consultado el 15 de abril de 2012.
4. CORTÉS, U. Comunicación entre Agentes Autónomos. Disponible en: <http://www.lsi.upc.es/~ia/ComunicacionA2006.ppt>. Consultado Octubre de 2011.
5. CRUTZEN, R.; PETERS, G.; DIAS, S.; FISSER, E.; GROLLEMAN, J. An Artificially Intelligent Chat Agent That Answers Adolescents' Questions Related to Sex, Drugs, and Alcohol: An Exploratory Study. In: Journal of Adolescent Health, Vol. 48, No. 5. (2011); pp. 514-519, Disponible en: <http://www.sciencedirect.com/science/article/pii/S1054139X10004301>. Consultado en Abril de 2012.
6. EDWARDS, T.; SANKARANARAYANAN, S. Applications of Intelligent Agents in Hospital Search and Appointment System. International Journal of E-Services and Mobile Applications IJESMA. Vol. 3, N.4. (2011);pp. 57-81.

7. GOMEZ-SANZ, J.; FUENTES, R. The INGENIAS Methodology. Fourth Iberoamerican Workshop on Multi-Agent Systems Iberagents. 2002.
8. IGLESIAS, C. Definición de una metodología para el desarrollo de Sistemas Multi-Agente. Tesis doctoral. Departamento de ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid. 1998. Disponible en: <http://www.upv.es/sma/teoria/agentes/tesiscif.pdf>. Consultado en Abril de 2012.
9. JADE. Java Agent DEvelopment Framework, Disponible en: <http://jade.tilab.com>. Consultado en Abril de 2012.
10. JARA, J.; CHACON, M.; ZELAYA, G. Empirical evaluation of three machine learning method for automatic classification of neoplastic diagnoses. Revista Chilena de Ingeniería. Vol. 19 N° 3. (2011); pp. 359-368.
Disponible en: <http://www.scielo.cl/pdf/ingeniare/v19n3/art06.pdf>. Consultado en Abril de 2012.
11. JIMÉNEZ, A. Ontologías para comunicación entre agentes, Disponible en: <http://alfonsojimenez.com/computers/ontologias-para-comunicacion-entre-agentes/>. Consultado el 26 de marzo de 2011.
12. KINNY, D.; GEORGEFF, M.; RAO, A. A design methodology for BDI agent systems. Technical Report 54, Australian Artificial Intelligence Institute, Melbourne, Australia. 1995.
13. LASHENG, Y.; JIE, L.; BEIJI, Z. Research on a Multi-agent Based Integrated Health Monitoring System for the Elderly at Home. En: Future Wireless Networks and Information Systems, Editorial: Springer Berlin Heidelberg. Vol. 143. (2012) pp. 155 – 161.
Disponible en: http://dx.doi.org/10.1007/978-3-642-27323-0_20. Consultado en Abril de 2012.
14. MARCHETTI, T.; GARCÍA, A. Evaluación de plataformas para el desarrollo de sistemas multiagente, en [http://cs.uns.edu.ar/~ajg/papers/2003\(Cacic\)MarchettiGarcia.pdf](http://cs.uns.edu.ar/~ajg/papers/2003(Cacic)MarchettiGarcia.pdf). 2003. Consultado febrero de 2012.
15. MATURANA, S.; FERRER, J.; BARANAO, F. Design and implementation of an optimization-based decision support system generator. European Journal of Operational Research, Volume 154, Issue 1, 1 April 2004, Pages 170-183. 2004.
16. MOLINA J.; GARCÍA, J. y BERNARDOS, A. Agentes y sistemas multiagente, Disponible en: www.ceditec.etsit.upm.es/index.php/component?option=com_docman/task,doc_download/gid,3/Itemid,78/. Consultado Mayo de 2011.
17. MORENO, F.; OÑA A. y MARTÍNEZ M. Un sistema de simulación como alternativa en el entrenamiento de habilidades deportivas abiertas. En: Revista de ciencias de la actividad física y del deporte. N° 4. pp. 75-95. 1998.
18. ROMEO, L.; PASTOR, M. Simulación del comportamiento humano. En: Revista del Instituto Nacional de Seguridad e Higiene en el Trabajo. N° 31. 2004.
19. SANORES, C.; PAVÓN, J. Simulación social basada en agentes. En: Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial. Vol. 9. N. 25. (2005);pp. 71-78
20. SCOTT, A.; DE LOACH, M.; WOOD, F. and SPARKMAN, H. Multiagent Systems Engineering. In: International Journal of Software Engineering and Knowledge Engineering. Vol. 11, No. 3 (2001.);pp. 231-258
21. SHOHAM, Y. Agent-oriented programming. En: Journal Artificial Intelligence. Vol. 60 pp. 51-92. 1993.
22. UNIVERSIDAD POLITÉCNICA DE MADRID. Grupo de investigación en sistemas inteligentes. Informe Técnico: Arquitecturas de Agente. Proyecto DAMMAD: Diseño y Aplicación de Modelos Multiagente para Ayuda a la Decisión (TIC2000-1370-04-01). Disponible en: http://platon.esct.urjc.es/grupo/proyectos/dammad/propio/Documentos/Informes/coord_survey.pdf. Consultado en abril de 2012.
23. UNAL. Universidad Nacional de Colombia. Inteligencia Artificial Distribuida, Disponible en: http://www.virtual.unal.edu.co/cursos/ingenieria/2001394/docs_curso/contenido.html. Consultado Abril de 2012.
24. WOOLDRIDGE, M.; JENNINGS, N. Intelligent Agents: Theory and practice. En: The Knowledge Engineering Review. Vol. 10:2. pp. 115-152. 1995.
25. WOOLDRIDGE, M.; JENNINGS, N.; KINNY, D. The Gaia Methodology for Agent-Oriented Analysis and Design. En: Journal of Autonomous Agents and Multi-Agent Systems. Vol. 3 No. 3. pp. 285-312. 2000.
26. WOOLDRIDGE M. An Introduction to Multiagent Systems. Second Edition, Wiley & Sons. 366 p. 2009.
27. WOOLDRIDGE, M; JENNINGS, N. Agent Theories, Architectures, and Languages: A Survey. University of Southampton. Disponible en <http://eprints.ecs.soton.ac.uk/2177/1/ECA194-WS.pdf>. Consultado en Abril de 2012.