

Universidad “José Martí”
Facultad de Informática
Carrera de Ingeniería Informática



Trabajo de Diploma para optar por el título de Ingeniería Informática.

**Título: Sistema Multiagente para revertir la emoción negativa tristeza
presente en los estudiantes durante su interacción con un ambiente de
enseñanza aprendizaje virtual**

Autora: Lisandra Torres Ruano.

Tutora: MSc. Irina Machado Mutis.

Sancti Spíritus, Cuba

Curso 2015- 2016

Pensamiento

Como seres humanos, todos queremos ser felices y estar libres de la desgracia, todos hemos aprendido que la llave de la felicidad es la paz interna. Los mayores obstáculos para la paz interna son las emociones perturbadoras como el odio, apego, miedo y suspicacia, mientras que el amor y la compasión son las fuentes de la paz y la felicidad.

Dalai Lama.

Agradecimientos

A mi madre por su amor incondicional y su ayuda sin importar los motivos.

A mi hermana por su ayuda y comprensión en todo momento.

A mis abuelos por darme tanto ánimo para continuar con esta difícil tarea.

A mi tutora la MSc. Irina Machado Mutis por su apoyo y colaboración constante.

A la Dr. Lidia Ríos por ser un ángel.

A mis amigos Maday y Bonet por estar presentes en las buenas y malas durante toda la universidad y en la realización de esta investigación, donde tanto hemos sufrido pero también hemos tenido muchas alegrías.

Dedicatoria

A mi madre y hermana por su constante apoyo incondicional

Resumen

En la Universidad de Sancti Spíritus actualmente se ha abierto una investigación en el departamento de Ingeniería Informática que trae como objetivo la creación de un sistema multiagente para revertir las emociones negativas presentes en los estudiantes que se pueda integrar con un ambiente de enseñanza aprendizaje virtual como APA-Prolog. Para la confección de este sistema se ha seleccionado la metodología PASSI y como plataforma de desarrollo JADE, teniendo en cuenta que se implementará completamente en Java, utilizando el Netbeans como IDE. Para ello se utiliza agentes, que unidos conforman sistemas multiagentes, los cuales dentro del campo de la computación afectiva, que no es más que el uso de computadoras o dispositivos para el tratamiento de emociones, constituyendo un tema novedoso, posibilitando conocer su amplitud y la importancia de estos aplicado al manejo de las emociones en los estudiantes durante el proceso de enseñanza. La plataforma JADE posibilita el manejo y registro de todos los agentes, asegurando su existencia durante todo el proceso y su futura utilización, esta plataforma es vinculada al Netbeans, el cual es utilizado para la implementación del agente completamente en Java teniendo en cuenta que toda la estructura del agente será implementada utilizando sus propias librerías, sin tener dependencia del Netbeans. Este sistema genera interfaces que serán las encargadas de interactuar con el usuario permitiéndole inicializarse en este.

Abstract

In the Sancti Spiritus University actually has opened an investigation in the engineering department who brings as goal the creation of multiagent system to revert the negatives emotion in the students to be able to integrate with an environment of virtual education apprenticeship like APA-Prolog. To the confection of this system has been selected the PASSI methodology and JADE as platform of development, it has been implemented completely in Java, using the Netbeans as IDE. To the confection of this program are used the agent that united create a multiagent system, which inside of the affective computer field constituted a newfangled theme facilitating to know its amplitude and importance aplicated to the emotion's management in the student during the process of education. The JADE platform allows the management and enrollment of all the agents. This platform is linked to the Netbeans that it is used to the implementation of the agente totally in Java and it will be independent. This system generate interfaces that will interact with the user allow it to start the program.

Contenido

Introducción	11
Capítulo 1: Sistema Multiagente y sus herramientas.....	18
Introducción	18
1.1-Programación orientada a agentes	18
1.2-Agentes	19
1.3-Sistemas Multiagentes	22
1.4- Metodologías para el desarrollo de Sistemas Multiagentes.....	23
1.4.1-Metodologías de desarrollo de agentes	23
PASSI.....	24
1.4.2- Plataformas de desarrollo de agentes.....	26
FIPA-OS.....	27
JADE	27
1.5-Lenguaje de programación	29
1.6-Netbeans	30
1.7- Lenguaje de Consulta Estructurado	30
1.8-Base de Datos.....	31
1.9-Tratamiento de la Tristeza.....	32
Conclusiones.....	33
Capítulo 2: Diseño del SMA utilizando PASSI	34
Introducción	34
2.1- Modelo de Requerimientos del Sistema	34
2.1.1-Descripción del Entorno.....	34
2.1.2- Descripción del Dominio	35
2.1.3-Identificación de Agentes.....	38
2.1.4-Identificación de Roles.....	40
2.1.5-Especificación de Tareas	44
2.2-Modelo de Sociedades de Agentes.....	46
2.2.1-Descripción de Ontología	46
2.2.1.1-Diagrama de Descripción de Ontología de Dominio	46
2.2.1.2-Diagrama de Descripción de Ontología de Comunicación.....	47
2.2.2-Descripción de Roles	47
2.2.3-Descripción de Protocolos	48
Conclusiones.....	49

Capítulo 3: Modelos de Implementación de Agentes	50
Introducción	50
3.1-Modelo de Implementación de agente.....	50
3.1.1-Definición de estructura del SMA.....	50
3.1.2-Definición de estructura del agente simple.....	51
3.1.3-Descripción de conducta del agente.....	53
3.1.3.1-Descripción de conducta del SMA.....	53
3.1.3.2-Descripción de conducta del Agente simple.....	54
3.2-Modelo de Código.....	54
3.2.1-Biblioteca de código reutilizable.....	54
3.3-Modelo de Despliegue	57
3.3.1-Configuración de Despliegue	57
3.4-Interfaces del SMA	58
Conclusiones.....	61
Conclusiones.....	62
Recomendaciones	63
Bibliografía	64
Anexos	66

Índice de Figuras

Figura 1: Agente	20
Figura 2: Fases y Niveles de PASSI	25
Figura 3: Diagrama de Entorno	35
Figura 4: Diagrama de Descripción del Dominio	35
Figura 5: Diagrama de Identificación de Agentes	39
Figura 6: Diagrama de Secuencia. Escenario Gestionar Usuario.	40
Figura 7: Diagrama de Secuencia. Escenario Gestionar Emoción	41
Figura 8: Diagrama de Actividades: Agente Mensajero	45
Figura 9: Diagrama de Actividades: Agente Psicologo3	45
Figura 10: Diagrama de Descripción de Ontología del Dominio	46
Figura 11: Diagrama de Descripción de Ontología de Comunicación.	47
Figura 12: Diagrama de Descripción de Roles.	48
Figura 13: Diagrama de Descripción de Protocolos.	49
Figura 14: Diagrama Definición de Estructura del SMA.	51
Figura 15: Diagrama Definición de Estructura del Agente Mensajero.	52
Figura 16: Diagrama Definición de Estructura del Agente Psicologo3	52
Figura 17: Descripción de conducta del SMA: Gestionar Usuario	53
Figura 18: Descripción de conducta del SMA: Gestionar Emoción	53
Figura 19: Descripción de Conducta del Agente Psicologo3.	54
Figura 20: Diagrama de Despliegue.	57
Figura 21: Interfaz: Gestionar usuario.	58
Figura 22: Interfaz: Obtener emoción	58
Figura 23: Tratamientos de Tristeza: Fotos.	59
Figura 24: Tratamientos Tristeza: Música.	59
Figura 25: Tratamientos Tristeza: Fábulas	60
Figura 26: Tratamientos Tristeza: Frases	60

Índice de Tablas

Tabla 1: Especificación del Casos Uso Autenticar	36
Tabla 2: Especificación del Casos Uso Crear Usuario	36
Tabla 3: Especificación del Casos Uso Validar Información.	37
Tabla 4: Especificación del Casos Uso Actualizar Perfil.	37
Tabla 5: Especificación del Casos Uso Obtener Emoción.	37
Tabla 6: Especificación del Casos Uso Tratar Tristeza.	38
Tabla 7: Gestionar Usuario.	42
Tabla 8: Gestionar Emoción.	43

Introducción

Las emociones son responsables de nuestro estado de ánimo y responsable de un conjunto de reacciones y no son más que reacciones psicofisiológicas de las personas ante situaciones relevantes desde un punto de vista adaptativo, tales como aquellas que implican peligro, amenaza, daño, pérdida, éxito, novedad, etc.(JOSÉ PIQUERAS 2010). El grado de intensidad en el que se pueden manifestar depende del ánimo que presenta una persona, de acuerdo a su personalidad, su forma física, así como la situación en la se encuentre. Poseen gran importancia en el devenir diario de las personas, ya que su desempeño en la vida social y personal estará marcado por su estado de ánimo y la necesidad de equilibrarlo o mejorarlo de acuerdo a la emoción que abriguen.

Entre las emociones podemos distinguir al menos dos grupos: las positivas y las negativas.

- Las emociones positivas tales como: alegría, gratitud, serenidad, interés, esperanza y amor, son los estados de ánimo y formas de responder ante el medio que ayudan a llevar una vida más plena, feliz y productiva. Un estudio longitudinal de la Universidad de Harvard mostró que las emociones positivas correlacionaban con longevidad, espiritualidad, buena salud física y calidad en las relaciones interpersonales. Otros estudios comparativos de las emociones identificaron aproximadamente una emoción positiva por cada tres o cuatro emociones negativas.
- El segundo grupo está conformado por las emociones negativas como el miedo-ansiedad, la ira, la tristeza-depresión y el asco.

Como parte del grupo de las emociones negativas esta investigación esta investigación se centrará en la tristeza, la cual es adquirida y es un estado provocado por un decaimiento de la moral. Es la expresión del dolor afectivo, se expresa mediante el llanto, el rostro abatido, la falta de apetito, se puede presentar cuando nuestras expectativas no se ven cumplidas o por

circunstancias doloras de la vida. Es un reflejo subjetivo de la realidad objetiva. Esta provoca una disminución de la energía e ilusión por las actividades cotidianas, paraliza nuestro metabolismo y nos limita y aísla. En casos graves, podemos alcanzar la depresión.

En el tema de las emociones hace dos décadas se está realizando un estudio de lo que será dotar a las computadoras de la posibilidad de tratar las emociones presentes en los usuarios, que actualmente hoy en día es el mayor por ciento de la población mundial, ya que el avance de las tecnologías ha provocado que los ordenadores sean utilizadas en todas las esferas de la sociedad, debido a la facilidad que brindar a la hora de trabajar. A pesar del tiempo que se lleva estudiando el tema, no ha sido hasta hace relativamente poco tiempo que un equipo de investigadores del MIT (Massachusetts Institute of Technology) lo han bautizado con el nombre de computación afectiva.

La computación afectiva es una línea de estudio que engloba informática, psicología, estudios conductuales y ciencia cognitiva, donde su objetivo es que ordenadores, cámaras o dispositivos mecánicos puedan interpretar los comportamientos y gestos de los seres humanos adaptándose a ellos.

Tiene como una de sus principales aplicaciones la esfera de la enseñanza, donde las emociones forman parte importante de la vida psicológica del escolar y tienen una alta influencia en la motivación académica y en las estrategias cognitivas (adquisición, almacenamiento, recuperación de la información, etc.), y por ende en el aprendizaje y en el rendimiento escolar (PEKRUN 1992).

Por lo que modelar ciertos aspectos afectivos de los estudiantes puede contribuir a brindarles una mejor situación de aprendizaje. Todo esto conlleva a una sola pregunta: ¿cómo lograr que un sistema computacional realice una tarea de esta naturaleza?

Rosalind Picard (1997) clasifica los tipos de características que serían necesarias para proveer a las computadoras con los medios para tomar en cuenta los estados afectivos de los estudiantes como sigue:

1. Reconocer emociones: Este aspecto es considerado como la habilidad para inferir el estado emocional a partir de observaciones de las expresiones y por medio del razonamiento acerca de una situación generadora de emociones.
2. Expresar emociones: El requerimiento básico para que una computadora pueda expresar emociones es que tenga canales de comunicación como voz o imagen y la habilidad para comunicar información afectiva sobre esos canales.
3. Tener emociones: Para cumplir con esta característica, es necesario cubrir aspectos como tener emociones primarias (aquellas ligadas a respuestas innatas a eventos potencialmente dañinos como el enojo y el miedo); tener emociones generadas cognitivamente (que involucran el razonamiento cognitivo explícito en su generación).
4. Tener inteligencia emocional: Esta característica podría observarse en una computadora que no sólo supiera expresar y tener emociones, sino que también supiera cómo administrar su expresión y cómo utilizar sus emociones para pensamiento creativo y motivación.

Para ello primeramente se debe establecer cuáles serán las emociones a tratar por el sistema, con el objetivo de determinar en qué consisten cada una, así como su tratamiento y vincular este último hacia un enfoque informático.

A la hora de establecer las emociones que el sistema tendrá, se tiene en cuenta las emociones primarias, (Ira, Miedo, Amor, Tristeza), las cuales son emociones básicas que se dan en todos los individuos de las más diversas culturas, poseen un sustrato biológico considerable, son esencialmente agradables o desagradables, nos activan y forman parte de la comunicación con los demás y a su vez, pueden actuar como poderosos motivos de la conducta (Canon-Videl and Miguel-Tobal 2001).

En cuanto a la tristeza existe gran variabilidad cultural e incluso algunas culturas no poseen palabras para definirla. Existen investigadores que plantean que la tristeza-depresión, al igual que cualquier otra emoción, tiene una función filogenética adaptativa para recabar la atención y el cuidado de los demás, constituir un modo de comunicación en situaciones de pérdida o separación, o ser un modo de conservar “energía” para poder hacer frente a ulteriores procesos de adaptación (*Whybrow 1984*).

Actualmente ya se han desarrollado sistemas capaces de actuar de acuerdo a las emociones presentes, entre los cuales se pueden encontrar ejemplos como:

- ✓ Proyecto Emociones Software cuyo objetivo es el desarrollo de una aplicación para dispositivos táctiles que ayude a los niños con un Trastorno del Espectro del Autismo (TEA)(Mancilla 2013)con sus habilidades sociales y de empatía.
- ✓ Proyecto Affective, surgido del equipo de investigación del grupo de computación afectiva Massachusetts Institute of Technology (MIT-2014),este proyecto ha comercializado con éxito lo que denominan “tecnologías de la emoción”, o lo que es lo mismo, un sistema automatizado de la plataforma de codificación facial con un sensor biométrico portátil.
- ✓ Emotient, también se especializa en el reconocimiento facial, pero su principal objetivo es el sector minorista. Su software está a prueba en las tiendas, al identificar 44 movimientos faciales para monitorear las reacciones emocionales del personal y los compradores, así como información demográfica incluida la edad y el género.
- ✓ Un automóvil empático es el objetivo de AutoEmotive, un proyecto de investigación del grupo Affective Computing en el laboratorio de medios del MIT en Estados Unidos, que se enfoca en explorar el potencial de las conexiones emocionales con máquinas.

- ✓ El movimiento QuantifiedSelf (Yo Cuantificado), en este caso los miembros de la sección ubicada en Londres, Inglaterra crearon herramientas como MoodScope y Mappiness que ayudan al usuario a emparejar su estado mental con eventos externos.

En Cuba poco se ha investigado en este sentido sin embargo, en la Universidad de Sancti Spiritus "José Martí" (UNISS) desde hace algunos años se vienen realizando investigaciones en el campo de la Inteligencia Artificial (IA) y sus aplicaciones, obteniendo como resultado un medio de enseñanza-aprendizaje virtual llamado APA-Prolog (R. 2008), su interfaz está basada en un mapa conceptual que contiene la mayoría de los conceptos asociados a la Programación Lógica, en el cual se utilizan agentes inteligentes para la navegación asistida.

A pesar de ser un sistema adaptativo esta herramienta no tiene en cuenta las emociones que presentan los estudiantes en su interacción con este medio de enseñanza-aprendizaje virtual. Esto ha provocado que en ocasiones los estudiantes presenten tristeza y que el aprovechamiento de su tiempo frente a la máquina no sea productivo.

Teniendo en cuenta esta situación se realizó el diseño de un sistema multiagente por (Rojas 2014-2015) con el objetivo de revertir las emociones negativas presentes en los estudiantes universitarios, como la Ira, Tristeza y Miedo, pero este diseño no fue implementado y presentaba problemas por lo que esta investigación se plantea el siguiente problema.

¿Cómo revertir la tristeza como emoción negativa presente en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual?

Derivado del problema de investigación expuesto anteriormente se puede precisar que el objetivo genera se reduce a:

Desarrollar un Sistema Multiagente para revertir la emoción negativa tristeza presente en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual

Para poder cumplir con el objetivo se trazaron las siguientes **Preguntas de Investigación:**

1. ¿Cuáles son los fundamentos teóricos, metodológicos que permiten desarrollar un Sistema Multiagente para revertir la emoción negativa tristeza presente en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual?
2. ¿Cómo diseñar un Sistema Multiagente para revertir la emoción negativa tristeza presente en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual?
3. ¿Cómo implementar un Sistema Multiagente para revertir la emoción negativa tristeza presente en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual?

Posteriormente a esto se plantearon un conjunto de tareas de investigación que darán respuestas a las preguntas anteriormente establecidas, siendo estas:

1. Determinación de los fundamentos teóricos, metodológicos que permitan implementar un Sistema Multiagente para revertir la emoción negativa tristeza en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual.
2. Diseño de un Sistema Multiagente para revertir la emoción negativa tristeza presentes en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual.
3. Implementación de un Sistema Multiagente para revertir la emoción negativa tristeza presente en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual.

Para desarrollar cada uno de estos objetivos se dispuso este documento escrito en 3 capítulos:

- En el Capítulo 1: se define conceptos esenciales para la creación del sistema, tales como el de agentes, sistemas multiagentes y a su vez todas las metodologías y plataformas utilizadas en su confección.
- En el Capítulo 2: Se realiza el diseño del sistema multiagente, contemplando solo los dos primeros niveles del ciclo de vida de PASSI, en las cuales se realiza un primer acercamiento a todas las funcionalidades que interviene en el sistema y detalla los pasos a seguir en cada una.
- En el Capítulo 3: Se realizan los últimos niveles del diseño con PASSI, generando un código, que establece una base para la implementación del sistema. Se muestran las interfaces que conformarán el sistema ya elaborado.

Capítulo 1: Sistema Multiagente y sus herramientas.

Introducción

Como parte del estudio de la computación afectiva ligada a un proceso de enseñanza, se puede utilizar diferentes aplicaciones que permitan cumplir con el objetivo trazado. En el marco de IA, la cual plantea “Desarrollar sistemas que piensen y actúen racionalmente” (V. Julian may-jun. 2000), ha surgido un nuevo paradigma, conocido como «paradigma de agentes», tomando un gran auge entre los investigadores y el cual ha sido elegido para el tratamiento de las emociones, específicamente la tristeza. El proceso de creación del agente mediante herramientas informáticas, sus características y ventajas que posibilitarán en un futuro la utilización de un software capaz de tratar las emociones presentes en los estudiantes, posibilitando un alto nivel de aprendizaje.

Este capítulo estará encaminado a definir conceptos, tales como el de agente, estableciendo sus principales características, donde la unión de estos conformará un sistema multiagente, para el cual se trazan diferentes metodologías para su diseño, así como variadas plataformas de desarrollo, donde se implementará posteriormente el sistema teniendo en cuenta también el lenguaje de programación utilizado y la base de datos donde se almacena toda la información necesaria.

1.1-Programación orientada a agentes

La Programación Orientada a Agentes (POA) está motivada por la necesidad que existe de modelar sistemas complejos y de manejar arquitecturas abiertas que cambien continuamente y evolucionen para acomodarse a nuevos componentes, logrando así satisfacer nuevos requerimientos. La POA se

encuentra muy relacionada con la Ingeniería de software orientada a Agentes (ISOA) dados los propósitos de esta.

Esta ingeniería está siendo descrita como un nuevo paradigma para el área de investigación en Ingeniería de Software. Pero para que pueda convertirse en un nuevo paradigma aceptado por esta industria, se deben desarrollar herramientas y metodologías robustas y fáciles de usar(Tveit Mayo 2001). La aproximación Orientada a agentes promete la habilidad de construir sistemas flexibles con comportamientos complejos y sofisticados al combinar componentes modulares (los agentes)(Caire G. 2001). Estos sistemas estarían compuestos por agentes inteligentes, capaces de brindar rapidez y facilidades en la sociedad actual.

1.2-Agentes

El origen por el cual la tecnología de los agentes surge es por la necesidad de aplicarlos a los sistemas complejos, donde la aplicación de las técnicas existentes ha fracasado o es muy difícil de comprender o mantener.

Varios investigadores y grupos de investigación han definido el término de agente desde diferentes puntos de vista, esto ha influido a que en la actualidad existan diferentes definiciones de lo que es un agente, como por ejemplo:

- Hyacinth S. Nwana definió los agentes como“un componente software y/o hardware que es capaz de actuar para llevar a cabo tareas en lugar de un usuario”(Nwana 1996)
- Por otra parte FIPA (FoundationforIntelligentPhysicalAgents) proporciona la siguiente definición del término agente:”es una entidad que reside en entornos donde interpreta datos que reflejan eventos y ejecuta comandos que producen efectos en ese entorno”(FIPA 2002)

La dificultad se debe especialmente a que los agentes se pueden emplear en muchos dominios de aplicación, llevando consigo a que cada dominio asocie varios y diferentes atributos a la definición de agente.(Ver Figura1).

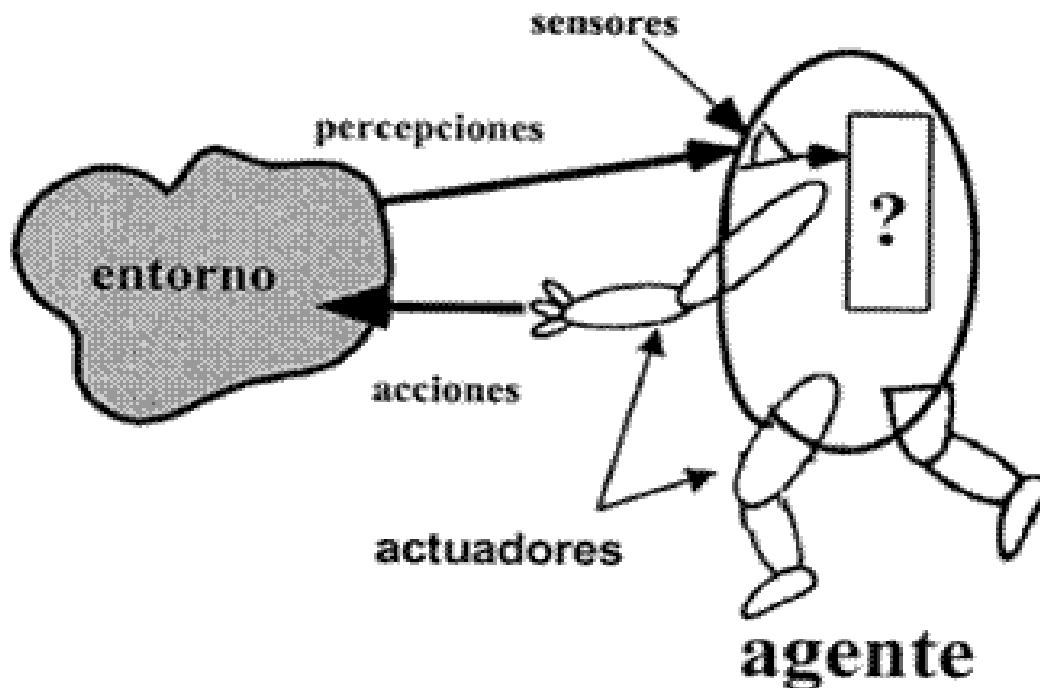


Figura 1: Agente

Tomado de (Pinto junio,2006)

Estos agentes están identificados por una serie de características, que deben cumplir, como:

- **Continuidad Temporal:** se considera un agente un proceso sin fin, ejecutándose continuamente y desarrollando su función.
- **Autonomía:** un agente es completamente autónomo si es capaz de actuar basándose en su experiencia. El agente es capaz de adaptarse aunque el entorno cambie severamente. Por otra parte, una definición menos estricta de autonomía sería cuando el agente percibe el entorno.

- **Sociabilidad:** este atributo permite a un agente comunicar con otros agentes o incluso con otras entidades.
- **Racionalidad:** el agente siempre realiza «lo correcto» a partir de los datos que percibe del entorno.
- **Reactividad:** un agente actúa como resultado de cambios en su entorno. En este caso, un agente percibe el entorno y esos cambios dirigen el comportamiento del agente.
- **Pro-actividad:** un agente es pro-activo cuando es capaz de controlar sus propios objetivos a pesar de cambios en el entorno.
- **Adaptatividad:** está relacionado con el aprendizaje que un agente es capaz de realizar y si puede cambiar su comportamiento basándose en ese aprendizaje.
- **Movilidad:** capacidad de un agente de trasladarse a través de una red telemática.
- **Veracidad:** asunción de que un agente no comunica información falsa a propósito.
- **Benevolencia:** asunción de que un agente está dispuesto a ayudar a otros agentes si esto no entra en conflicto con sus propios objetivos.

Hasta el momento no existe un consenso sobre el grado de importancia de cada una de estas propiedades para un agente, sin embargo, se puede afirmar que estas propiedades son las que distinguen a los agentes de meros programas. Según se ha visto en la definición de Wooldridge, las características de autonomía, reactividad, pro-actividad y sociabilidad son las características básicas.

En este caso quizás la definición tomada como referencia, siendo quizás la más simple es la de Russell, que considera “un agente como una entidad que percibe y actúa sobre un entorno”.(Russell 1996).

En la mayoría de las ocasiones, los agentes no son desarrollados de forma independiente sino como entidades que constituyen un sistema. A este sistema se le denomina **multi-agente (Huhns 1998)**

1.3-Sistemas Multiagentes

Los sistemas multiagentes son aquellos que se diseñan e implementan pensando en que estarán compuesto por varios agentes que interactuarán entre sí, de forma que juntos permitan alcanzar la funcionalidad deseada. En este caso se exige un esfuerzo de abstracción en el diseño, identificar mecanismos de aprendizaje, coordinación, negociación, etc.

Existen varias definiciones, como por ejemplo: un Sistema Multiagente se conoce por un conjunto de agentes autónomos, generalmente heterogéneos y potencialmente independientes, que trabajan en común resolviendo un problema. Estos agentes vinculados con la noción de agente inteligente son capaces de tomar la iniciativa, compartir conocimiento, cooperar y negociar, y comprometerse con metas comunes (Rodríguez Elias mayo,2005)

Las interacciones más habituales como son informar o consultar a otros agentes permiten a los agentes «hablar» entre ellos, tener en cuenta lo que realiza cada uno de ellos y razonar acerca del papel jugado por los diferentes agentes que constituyen el sistema.

Para el desarrollo de estos sistemas multiagentes se emplean una serie de metodologías y herramientas, las cuales pueden ser variadas

1.4- Metodologías para el desarrollo de Sistemas Multiagentes

Toda construcción de un sistema multiagente es un proceso minucioso, donde primeramente se debe tener en cuenta las funcionalidades, para luego pasar a un diseño detallado, puntualizando los componentes de este agente para poder transitar a la etapa final donde se lleva a cabo la implementación. Este proceso debe ser guiado por el uso de una metodología.

1.4.1- Metodologías de desarrollo de agentes

Al igual que la ingeniería del software tradicional, el éxito en la aplicación de la ingeniería del software orientada a agentes está relacionada con la correcta aplicación de una metodología y un proceso de desarrollo.

Actualmente no existe un consenso general en cuanto a cuál metodología es la correcta a utilizar en la creación de un agente, ya que la falta de herramientas adecuadas en la Ingeniería de software orientada a agentes ha traído como consecuencia la aparición de variadas metodologías, tales como: Gaia, MaSE, MESSAGE, Tropos, SODA, PASSI, Prometheus o INGENIAS.

- Gaia es una metodología que no ha evolucionado mucho a lo largo del tiempo trayendo como consecuencia que solo dos versiones se hayan elaborado. Se compone de dos fases: análisis y diseño, aunque la etapa de diseño suele dividirse a su vez en dos: diseño arquitectónico y diseño detallado. En consecuencia, la metodología no cubre la especificación de requisitos ni la fase de implementación. Se observa que los modelos de Gaia no tienen como objetivo ajustarse a ningún meta modelo específico, utilizando en su lugar una notación bastante informal, incluso dejándola a gusto del diseñador. Se puede decir que resultaría esperable la existencia de mayor rigor por parte de una metodología. (Zambonelli 2005).

- Tropos es una metodología de desarrollo orientada a agentes con larga trayectoria. La principal característica que le identifica respecto de otras alternativas, consiste en un mayor énfasis dirigido a la identificación y análisis de requisitos. Se compone de cinco fases: análisis de requisitos iniciales, análisis de requisitos tardíos, diseño arquitectónico, diseño detallado, y finalmente la fase de implementación. (Bresciani 2004).

PASSI

PASSI (Process for Agents Societies Specification and Implementation) integra modelos y conceptos orientados a objetos con ISOA, proponiendo una metodología "paso a paso" que cubre todas las fases del desarrollo (desde la especificación de requisitos hasta la implementación), caracterizada por promover la adhesión a los estándares siempre que sea posible. Siguiendo esta línea, considera el uso de tecnologías bien establecidas como UML como lenguaje de modelado, la plataforma de la FIPA como destino de la implementación y XML para estructurar las comunicaciones. (Cossentino 2005)

Sus desarrolladores le atribuyen gran importancia a la reutilización de patrones de agentes existentes; de ahí que haya sido concebida para ser soportada por una herramienta CASE, el PASSI Tool-Kit (PTK)

Esta herramienta constituye un plugg-in de Rational Rose (herramienta CASE (Computer Assisted Software Engineering) que integra todos los elementos que propone RUP (Proceso Unificado de Rational) para cubrir el ciclo de vida de un proyecto), que permite la división lógica de los modelos generados por cada una de las doce fases de la metodología en los niveles a que estas pertenecen. De igual forma, el PTK interviene en la generación automática de un gran número de estos modelos. (Pinto junio, 2006)

PASSI considera un proceso de desarrollo compuesto de cinco fases. Cada una de estas fases guía una producción de modelos, donde el resultado de cada fase será su modelo respectivo. Estos modelos son: Modelo de

Requerimientos del Sistema, Modelo de Sociedades de Agente, Modelo de Implementación de Agente, Modelo de Código y finalmente Modelo de Despliegue. (Ver Figura 2).

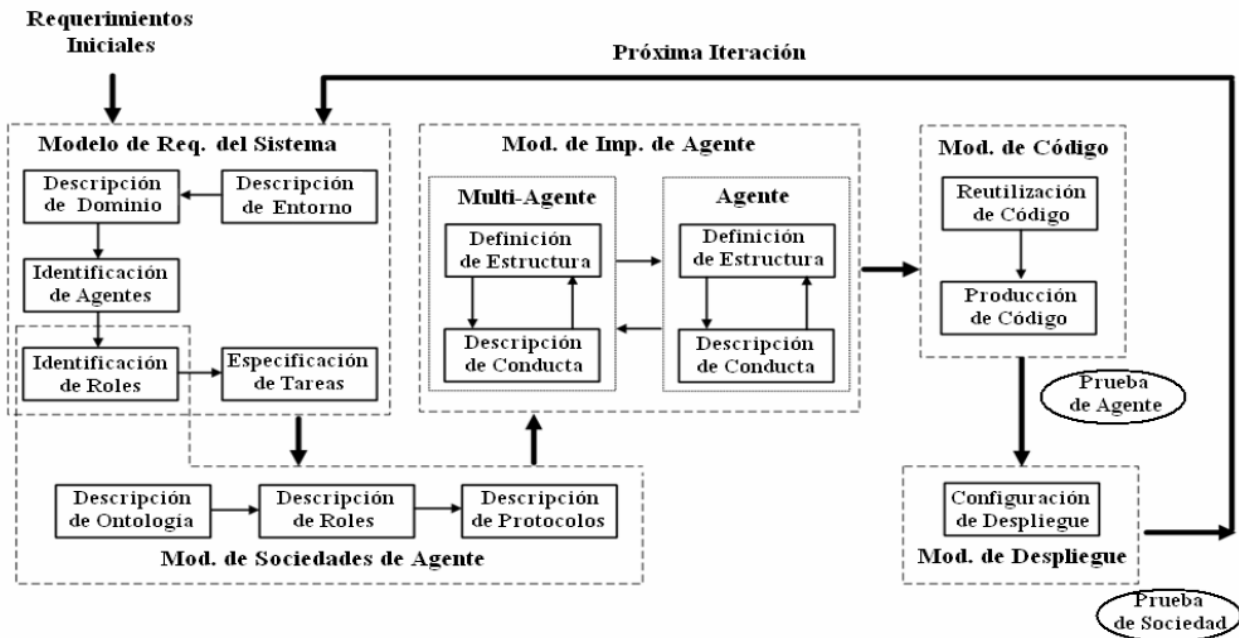


Figura 2: Fases y Niveles de PASSI

Fuente: (Pinto junio,2006)

El Modelo de Requerimientos del Sistema reúne los requisitos del sistema en cuatro etapas. Su primera etapa, Descripción de Requerimientos del Dominio, recoge en diagramas de casos de uso convencionales los requisitos de los escenarios posibles del sistema. Las demás etapas describen una especificación de alto nivel de los agentes, primero haciendo una identificación preliminar de los agentes y repartiendo los casos de uso, posteriormente planteando los roles de los agentes mediante diagramas de secuencia donde se presentan los escenarios principales del sistema, y finalmente se especifican las tareas asignadas a cada agente mediante diagramas de actividades.

El Modelo de Sociedades de Agente describe el conocimiento y las comunicaciones entre los agentes que pueblan el sistema y por tanto, la ontología del mismo. De esta forma, en las diferentes etapas, se compone una descripción de la ontología mediante un diagrama de clases, al que se añade

otro que modela las interacciones (una clase por cada clase de agente identificada y una relación por cada comunicación). Posteriormente se asignan roles a las clases de agentes identificadas mediante otro diagrama de clases (en este caso, los agentes se representan con paquetes y los roles con clases). Finalmente, se especifican los protocolos que guían las comunicaciones entre agentes, siguiendo los contenidos en el estándar FIPA.

El Modelo de Implementación de Agente comienza definiendo tanto la estructura general del sistema como la estructura individual de los agentes. En el primer caso se hará mediante un diagrama de clases que representa las interacciones significativas con el entorno mediante actores. Por otra parte, para la estructura individual, es suficiente con un diagrama de clase corriente por cada clase de agentes.

Para el diseño de este sistema multiagente se utiliza PASSI, ya que constituye un modo razonado de desarrollo que permite el tránsito de requerimientos a código para diseñar e implementar SMA. La misma opera sobre la base de integración de Modelos de Diseño y conceptos de Ingeniería de Software Orientados a Objeto lo cual unido a las tendencias actuales de la Inteligencia Artificial, en apoyo con el nivel de especificación que brinda el lenguaje de modelación UML, le atribuye un nivel de descripción acertado del objeto de aplicación. (PASSI 2003)

Esta metodología es compatible con la plataforma de desarrollo JADE, la que posibilitará la creación de un agente, así como la dirección de sus principales funcionalidades y comunicación.

1.4.2- Plataformas de desarrollo de agentes

El objetivo principal de una plataforma de desarrollo es la construcción de un sistema multiagente, pero en este caso las utilizadas por PASSI son: JADE y FIPA-OS, varias de estas plataformas han sido implementada en Java, pero teniendo en cuenta la metodología a utilizar solo se tiene en cuenta JADE.

FIPA-OS

FIPA-OS está diseñada para dar soporte a los estándares de agentes FIPA, la distribución de esta incluye soportes para:

- Diferentes tipos de AgentShells para producir agentes que puedan comunicarse con cada uno de ellos utilizando las facilidades que brinda FIPA-OS.
- Soportes de múltiples capas para la comunicación de agentes.
- Administración de conversaciones y mensajes.
- Configuración dinámica de la plataforma para soportar múltiples tipos de persistencia y múltiples codificaciones.
- Interfaces abstractas y modelo de diseño de software.
- Herramientas de visualización y diseño.

La distribución de FIPA-OS contiene archivos de clases, código fuente de Java y documentación. También incluye simples agentes de prueba para acceder a la plataforma de servicios de agente y alguna visualización de software. La arquitectura de FIPA-OS puede ser concebida como un modelo no estricto por capas. (Stefan Poslad 2005)

JADE

JADE es un middleware desarrollado por TILAB para el desarrollo de un sistema multiagente basado en la arquitectura de comunicación “amigo a amigo”.

El ambiente puede evolucionar dinámicamente con pares, en JADE llamados agentes, que aparecen y desaparecen en el sistema de acuerdo a las necesidades y los requerimientos del ambiente de la aplicación.

JADE es íntegramente desarrollado en Java y es basado en los siguientes principios:

- Interoperatividad: Es obediente con las especificaciones FIPA, lo que trae como consecuencia que los agentes puedan actuar conjuntamente con otros agentes con la condición de que cumplan con los mismo estándares.
- Uniformidad y Portabilidad: JADE provee una homogeneidad de APIs que son independientes la red y de una versión de Java.
- Fácil de usar: La complejidad de un middleware está escondida detrás de un simple e intuitivo **set de APIs**(Aplicaciones de Programación de Interfaces).
- Adaptativa: los programadores no necesitan usar todas las condiciones establecidas por el middleware. Las que no son usadas no deben ser de conocimiento para los programadores, ni deben incurrir en ningún gasto computacional adicional.

Para el desarrollo de este sistema se la plataforma seleccionada ha sido JADE, ya que permite simplificar el desarrollo de aplicaciones, variadas compañías ya lo usan para sus aplicaciones en diferentes sectores tales como el turismo, donde los mejores rasgos de una aplicación elaborada con JADE son:

- Aplicaciones distribuidas compuestas por entidades autónomas.
- Negociación y Coordinación
- Pro-Actividad
- Aplicaciones multi-partes
- Interoperatividad
- Abertura
- Versatilidad
- Fácil de usar y aplicaciones móviles.

Para utilizar correctamente esta herramienta se deben tener sólidos conocimientos sobre el lenguaje de programación Java.

1.5-Lenguaje de programación

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

Al programar en Java no se parte de cero. Cualquier aplicación que se desarrolle se apoya en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java). (Javier García de Jalón 2000)

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar el paradigma de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Java constituye una herramienta independiente del tipo de CPU utilizada, siendo un tipo de código neutro, el cual se ejecuta sobre una máquina virtual. Es un lenguaje muy completo (de hecho se está convirtiendo en un macro-lenguaje: Java 1.0 tenía 12 paquetes; Java 1.1 tenía 23 y Java 1.2 tiene 59). En cierta forma casi todo depende de casi todo. Por ello, conviene aprenderlo de modo iterativo: primero una visión muy general, que se va refinando en sucesivas iteraciones. Una forma de hacerlo es empezar con un ejemplo completo en el que ya aparecen algunas de las características más importantes.

1.6-Netbeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.(Domínguez-Dorado 2006)

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándoles nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.(Domínguez-Dorado 2006)

1.7- Lenguaje de Consulta Estructurado

S.Q.L. significa lenguaje estructurado de consulta (*Structured Query Language*). Es un lenguaje estándar de cuarta generación que se utiliza para definir, gestionar y manipular la información contenida en una Base de Datos Relacional. Se trata de un lenguaje definido por el estándar ISO/ANSI SQL que utilizan los principales fabricantes de Sistemas de Gestión de Bases de Datos Relacionales (SGBDR). En los lenguajes procedimentales de tercera generación se deben especificar todos los pasos que hay que dar para conseguir el resultado. (Montero 2006)

Es un lenguaje sencillo y potente que se emplea para la gestión de la base de datos a distintos niveles de utilización: usuarios, programadores y administradores de la base de datos.

Todos los principales SGBDR incorporan un motor SQL en el Servidor de Base Datos, así como herramientas de cliente que permiten enviar comandos SQL para que sean procesadas por el motor del servidor. De esta forma, todas las tareas de gestión de la Base de Datos (BD) pueden realizarse utilizando sentencias SQL.

- Consultar datos de la Base de Datos.
- Insertar, modificar y borrar datos.
- Crear, modificar y borrar objetos de la Base de Datos.
- Controlar el acceso a la información.
- Garantizar la consistencia de los datos.

La Base de Datos elaborada mediante SQL está contenida en el Gestor de Base de datos SQLite.

1.8-Base de Datos

SQLite es una base de datos muy similar a la conocida Access del mundo Windows pero a diferencia de esta posee una serie de ventajas que la hacen interesante de aplicar (Aguiar diciembre,2006)

- Para comenzar es multiplataforma y cumple con los estándares (en su mayoría) SQL92 por lo que su sintaxis y forma de uso casi no posee curva de aprendizaje a los conocedores de SQL y sus MySQL, porque además como este último (implementación de mysql en php) sqlite también está contemplada en el tratamiento dinámico de php profusamente.
- Se puede usar en modo ventana de comandos (Shell) o embebido en aplicaciones de código (casi cualquier lenguaje de programación directamente o indirectamente).
- Es ideal para trabajar con volúmenes medianos o pequeños de información, de manera ágil y eficiente. Aunque sus diseñadores aducen que es posible manejar bases de datos de 2 terabytes sin mayores inconvenientes.

Una vez establecidas todos los aspectos necesarios para la confección de un sistema multiagente, se procede a establecer los tratamientos necesarios para revertir la tristeza, objetivo de esta investigación.

1.9-Tratamiento de la Tristeza

El objetivo principal a la hora de establecer un tratamiento desde el punto de vista psicológico consta de dos partes:

- Cambiar el objeto que produce el daño.
- Ayudar a entenderlo.

Primeramente se debe hacer un cuestionario en el cual sea posible obtener la mayor información respecto a que se debe la tristeza, todos los factores que influyen en ese estado y finalmente después de haber identificado todas las características del estado de ánimo, teniendo en cuenta de a qué se debe pues comenzar el tratamiento con objetivos como: elevar la moral, dar ánimo.

De acuerdo a las causas que generan tristeza, estas se pueden generalizar en tres tipos:

- Proyectos no realizados
- Perdida dolorosa
- Patología

Teniendo en cuenta los recursos de la computadora los tratamientos en el ámbito de la computación afectiva mediante la utilización de agentes inteligentes se establecen como:

- Dar mensajes afectivos y de compasión relacionados al tema.
- Fabulas que inspiren enseñanzas.
- Poner música que pueda relajar al usuario teniendo en cuenta sus gustos.
- Mostrar fotos que eleven el ánimo y que provoquen alegría.

Conclusiones

El estudio de las diferentes metodologías reveló que solo PASSI modela todas las etapas del ciclo de vida de un SMA, pero a pesar de que no incluye inteligencia, fue la seleccionada pues ofrece una guía detallada para el desarrollo del SMA y genera parte del código de la explicación.

JADE y el lenguaje Java fueron los seleccionados como IDE y lenguaje de programación pues existe una total correspondencia entre ellos lo que trae ventaja en la fase de implementación.

SQLite fue seleccionado porque ofrece la capacidad de trabajar con volúmenes grandes y medianos de información de manera ágil y fácil.

Capítulo 2: Diseño del SMA utilizando PASSI

Introducción

Partiendo de las herramientas seleccionadas en el capítulo anterior y de la metodología a utilizar, en este capítulo se pretende realizar el diseño del SMA a través de las dos primeras fases de PASSI, generando los diagramas correspondientes.

2.1- Modelo de Requerimientos del Sistema

Este modelo constituye una visión general de lo que será el SMA, donde se establecen los actores que estarán presentes en el sistema, así como todas las funcionalidades a implementar.

2.1.1-Descripción del Entorno

Este modelo constituye el máximo nivel de abstracción del sistema, donde se definen dos tipos de actores: internos y externos. Particularizando al problema en cuestión en este diagrama se definen dos actores, el estudiante que será el encargado de interactuar con el software y la PC constituye la interacción del sistema con un recurso externo donde estará toda la información necesaria. El Caso de Uso identificado fue Revertir Emociones Negativas. (Figura 3)

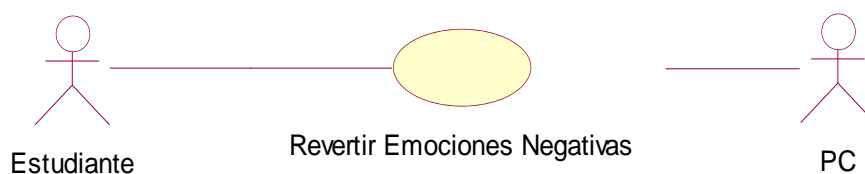


Figura 3: Diagrama de Entorno

2.1.2- Descripción del Dominio

Una vez establecido el diagrama de entorno, se procede a realizar como parte de la primera fase del diseño el diagrama de descripción del dominio. Este diagrama constituye un primer acercamiento a lo que será el sistema, mostrando de manera generalizada todas las funcionalidades. Compuesto por dos actores: el estudiante, quien inicializará el sistema y la PC que como recurso externo tendrá todos los tratamientos que aplicará, el resto de los elementos son las acciones que debe realizar el sistema. (Figura 4)

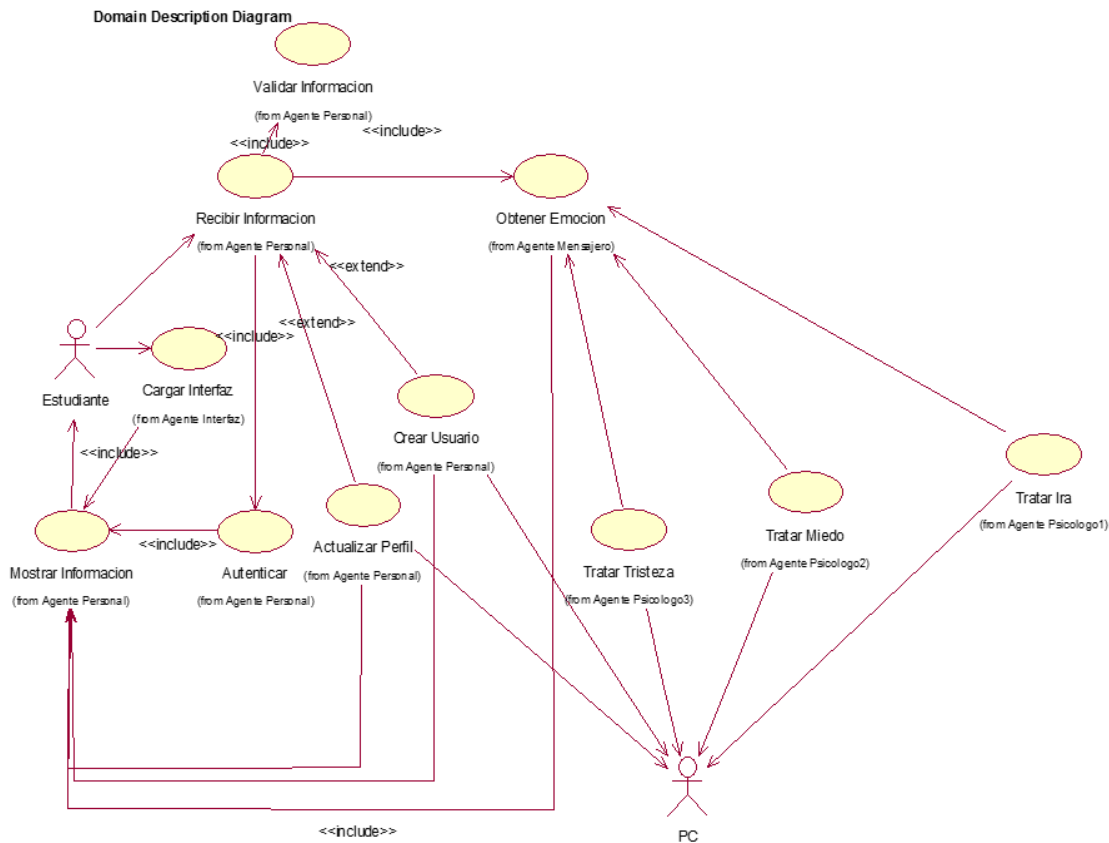


Figura 4: Diagrama de Descripción del Dominio

Los Casos de Uso identificados durante esta fase fueron los siguientes:

- Cargar Interfaz

- Mostrar Información
- Autenticar
- Recibir Información
- Validar Información
- Crear Usuario
- Actualizar Perfil
- Obtener Emoción
- Tratar Ira
- Tratar Miedo
- Tratar Tristeza

A continuación se realiza una breve descripción de los casos de uso por los que está constituido el sistema.

Tabla 1: Especificación del Casos Uso Autenticar

Caso Uso: Autenticar		
Descripción:	Este CU tiene como objetivo que el estudiante se autentique al iniciar su sesión con el sistema, se verificarán sus datos y se les permitirá el acceso en caso de que los datos estén bien, en caso contrario el sistema le solicitará que vuelva a autenticarse.	
Comunicaciones:	Iniciador	Participantes
	1. CU Autenticar.	1. CU Validar Información. 2. CU Recibir Información.

Tabla 2: Especificación del Casos Uso Crear Usuario

Caso Uso: Crear Usuario		
Descripción:	Este CU tiene como objetivo que el estudiante se registre en el software si nunca ha interactuado con él, y posteriormente gestione sus preferencias.	
Comunicaciones:	Iniciador	Participantes
	1. CU Crear Usuario.	1. CU Validar Información.

		2. CU Recibir Información.
--	--	----------------------------

Tabla 3: Especificación del Casos Uso Validar Información.

Caso Uso: Validar Información		
Descripción:	Al introducir los datos en el sistema este es el encargado de verificar que la información introducida sea correcta.	
Comunicaciones:	Iniciador	Participantes
	1. CU Autenticar.	1. CU Validar Información.

Tabla 4: Especificación del Casos Uso Actualizar Perfil.

Caso Uso: Actualizar Perfil		
Descripción:	Este CU tiene como objetivo permitir al estudiante cambiar algunos de sus datos como usuario, siendo el usuario el único dato imposible de modificar.	
Comunicaciones:	Iniciador	Participantes
	1. CU Actualizar Perfil	1. CU Recibir información 2. CU Validar Información.

Tabla 5: Especificación del Casos Uso Obtener Emoción.

Caso Uso: Obtener Emoción		
Descripción:	Este CU tiene como objetivo obtener la emoción seleccionada por el estudiante para poder comenzar su tratamiento.	
Comunicaciones:	Iniciador	Participantes
	1. CU Autenticar 2. CU Obtener emoción	1. CU Tratar Ira 2. CU Tratar Miedo. 3. CU Tratar Tristeza.

Tabla 6: Especificación del Casos Uso Tratar Tristeza.

Caso Uso: Tratar Tristeza		
Descripción:	Este CU tiene como objetivo tratar la tristeza que experimente el estudiante, estará compuesto por 4 tratamientos.	
Comunicaciones:	Iniciador	Participantes
	1. CU Obtener Emoción.	1. CU Tratar Tristeza.

2.1.3-Identificación de Agentes

Una vez definidos el entorno y el dominio se procede a la identificación de agentes. En este diagrama PASSI plantea la agrupación de CU por paquetes, organizados por agentes. (Figura 5)

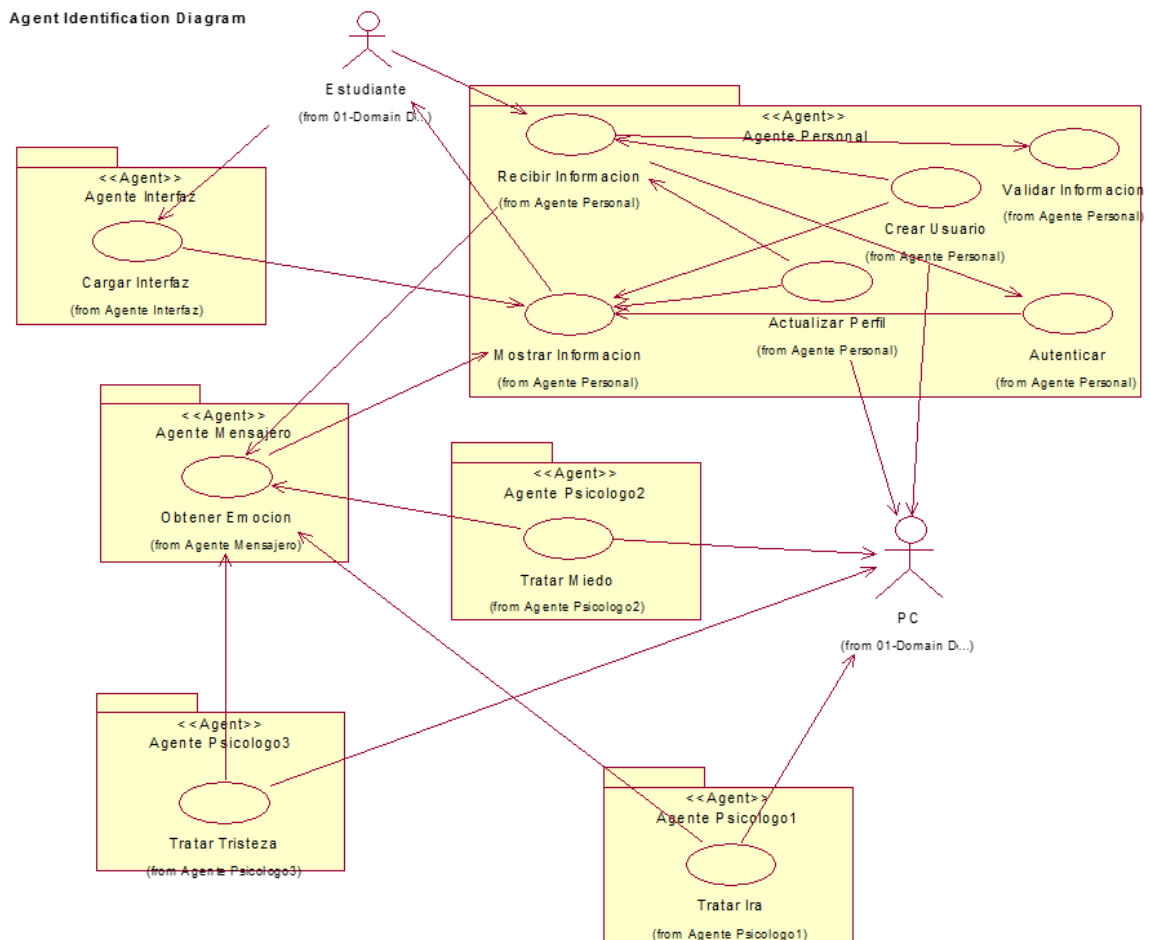


Figura 5: Diagrama de Identificación de Agentes

En este diagrama fueron identificados 6 agentes en el proceso de tratamiento de emociones negativas:

Agente Interfaz

1. CU Cargar Interfaz

Este agente será el encargado de una vez que el usuario inicie el software levantar la interfaz principal.

Agente Personal

1. CU Mostrar Información
2. CU Autenticar
3. CU Recibir Información
4. CU Validar Información
5. CU Crear usuario.
6. CU Actualizar perfil

Este agente será el encargado del manejo de todos los datos concernientes al usuario y de mostrar y recibir los datos y las confirmaciones.

Agente Mensajero

1. CU Obtener emoción

Este será el encargado de recibir la emoción seleccionada por el usuario y de acuerdo a esta, llamar a uno de los tres agentes encargados de los tratamientos.

Agente Psicologo1

1. CU Tratar Ira

Este será el encargado de dar tratamiento a la ira según las preferencias del estudiante y el tratamiento indicado para revertir dicha emoción.

Agente Psicologo2

1. CU Tratar Miedo

Este será el encargado de dar tratamiento al miedo según las preferencias del estudiante y el tratamiento indicado para revertir dicha emoción.

Agente Psicologo3

1. CU Tratar Tristeza

Este será el encargado de dar tratamiento a la tristeza según las preferencias del estudiante y el tratamiento indicado para revertir dicha emoción.

2.1.4-Identificación de Roles

Durante este nivel se identifican los roles de cada agente y se establecen paso a paso cada una de las funcionalidades de estos roles, con el objetivo de establecer el camino a seguir por el programador, se encuentra dividido en dos escenarios que generalizan todas las funcionalidades del sistema.

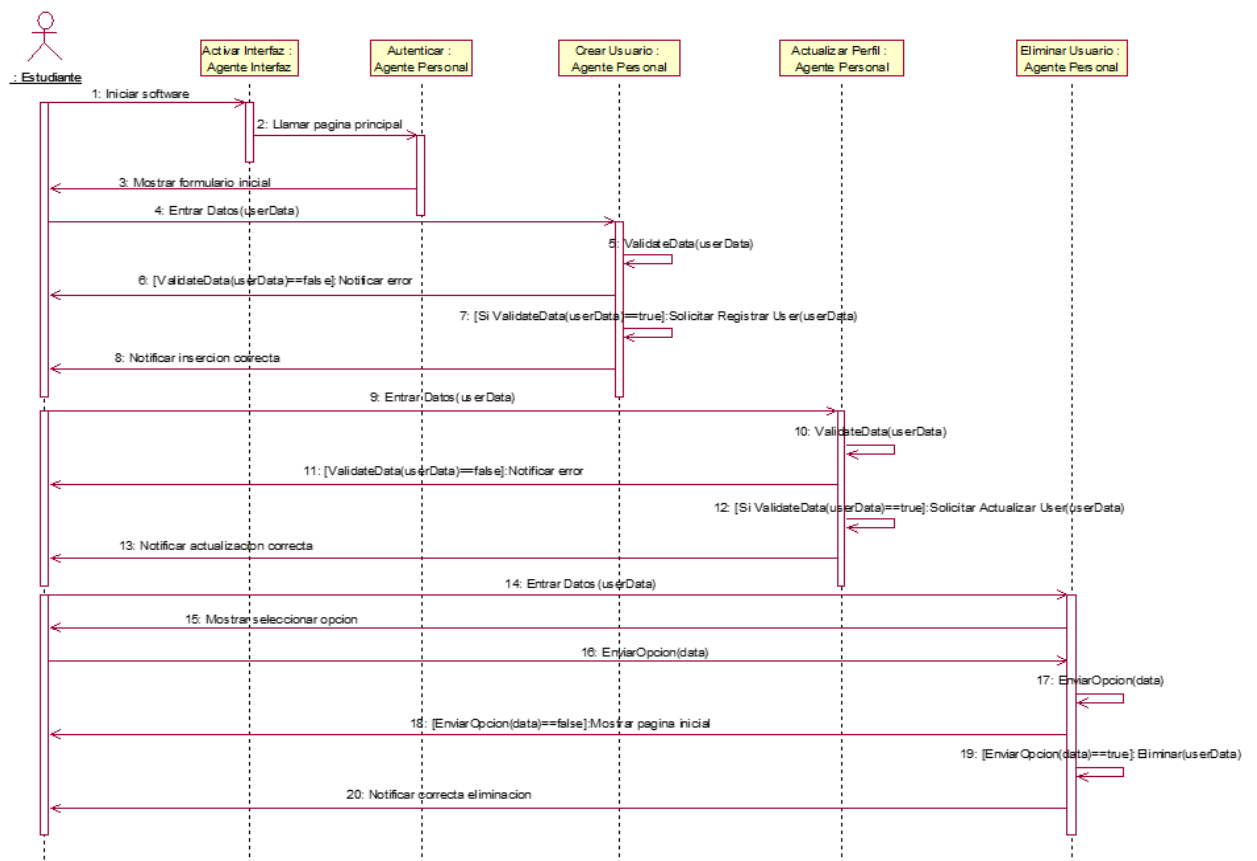


Figura 6: Diagrama de Secuencia. Escenario Gestionar Usuario.

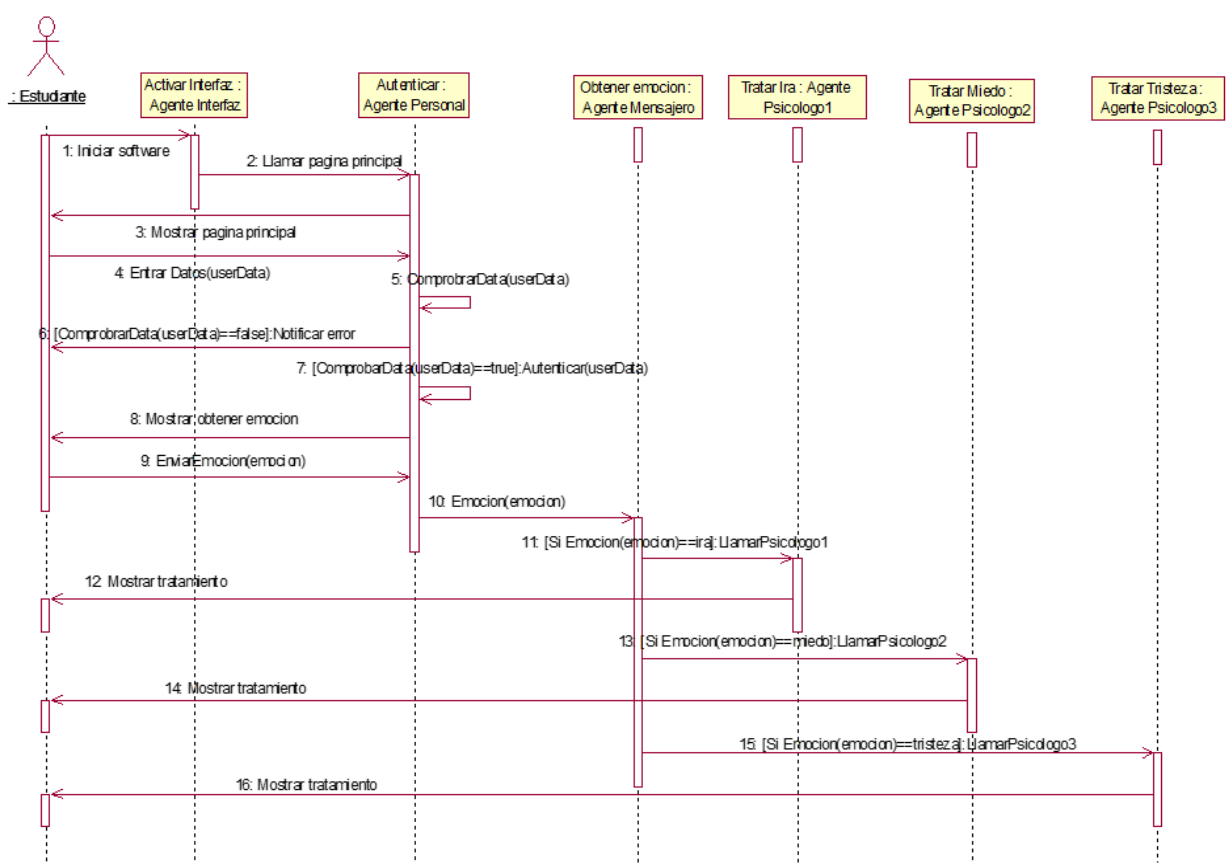


Figura 7: Diagrama de Secuencia. Escenario Gestionar Emoción

Identificación de Roles de los agentes propuestos

Agente Interfaz

1. Rol Activar Interfaz

Agente Personal

1. Rol Autenticar
2. Rol Crear Usuario
3. Rol Actualizar Perfil
4. Rol Eliminar Usuario

Agente Mensajero

1. Obtener emoción

Agente Psicologo1

1. Rol Tratar Ira

Agente Psicologo2

1. Rol Tratar Miedo

Agente Psicologo3

1. Rol Tratar Tristeza

Tabla 7: Gestionar Usuario.

Escenario: Gestionar Usuario	
Descripción	Este escenario muestra el curso normal que sigue un usuario externo para acceder al sistema.
CU Asociados:	CU Crear Usuario. CU Validar Información. CU Recibir Información. CU Actualizar Perfil
Roles Asociados	Descripción
Rol Activar Interfaz (Agente Interfaz)	Este rol es el responsable de la comunicación entre los usuarios y el sistema. Su principal actividades activar la interfaz con el agente personal
Rol Autenticar (Agente Personal)	Este rol es el responsable de posibilitar que el usuario acceda correctamente al sistema.
Rol Crear usuario (Agente Personal)	Este rol es el responsable de posibilitar que el usuario cree un perfil en el sistema para poder acceder a sus funcionalidades.
Rol Actualizar Perfil (Agente Personal)	Este rol es el responsable de posibilitar que el usuario acceda a sus datos y los modifique.
Rol Eliminar Perfil	Este rol es el responsable de posibilitar

(Agente Personal)	que el usuario se elimine del sistema.
--------------------------	--

Tabla 8: Gestionar Emoción.

Escenario: Gestionar Emoción.	
Descripción	Este escenario muestra el curso normal que sigue el sistema al obtener la emoción.
CU Asociados:	<ol style="list-style-type: none"> 1. CU Autenticar 2. CU Obtener Emoción. 3. CU Tratar Ira. 4. CU Tratar Tristeza. 5. CU Tratar Miedo.
Roles Asociados	Descripción
Rol Activar Interfaz (Agente Interfaz)	Este rol es el responsable de la comunicación entre los usuarios y el sistema. . Su principal actividades activar la interfaz con el agente personal
Rol Autenticar (Agente Personal)	El objetivo de este rol es permitir la entrada de los diversos usuarios registrados al sistema. Sus principales actividades son verificar su usuario y contraseña.
Rol Obtener Emoción (Agente Mensajero)	Este rol es el encargado de captar la emoción sentida por el usuario una vez que haya accedido al sistema. Su principal actividad es captar la emoción.
Rol Tratar Ira (Agente Psicólogo1)	Este rol es el encargado de tratar la Ira en caso de que esta sea la emoción sentida por el estudiante. Su principal

	actividad es tratar la ira.
Rol Tratar Miedo (Agente Psicólogo2)	Este rol es el encargado de tratar el miedo en caso de que esta sea la emoción sentida por el estudiante. Su principal actividad es tratar el miedo.
Rol Tratar Tristeza (Agente Psicólogo3)	Este rol es el encargado de tratar la tristeza en caso de que esta sea la emoción sentida por el estudiante. Su principal actividad es tratar la tristeza.

Una vez ya establecidos las funcionalidades a ejecutar el sistema, a partir de este momento esta investigación solo se referirá a los agentes tristeza y mensajero dentro del diseño del SMA.

2.1.5-Especificación de Tareas

En esta etapa se muestran los diagramas de actividades de los agentes: Agente Mensajero y Agente Psicologo3, describiendo mediante actividades todas las tareas de los agentes. (Ver Anexos 1 al 4)

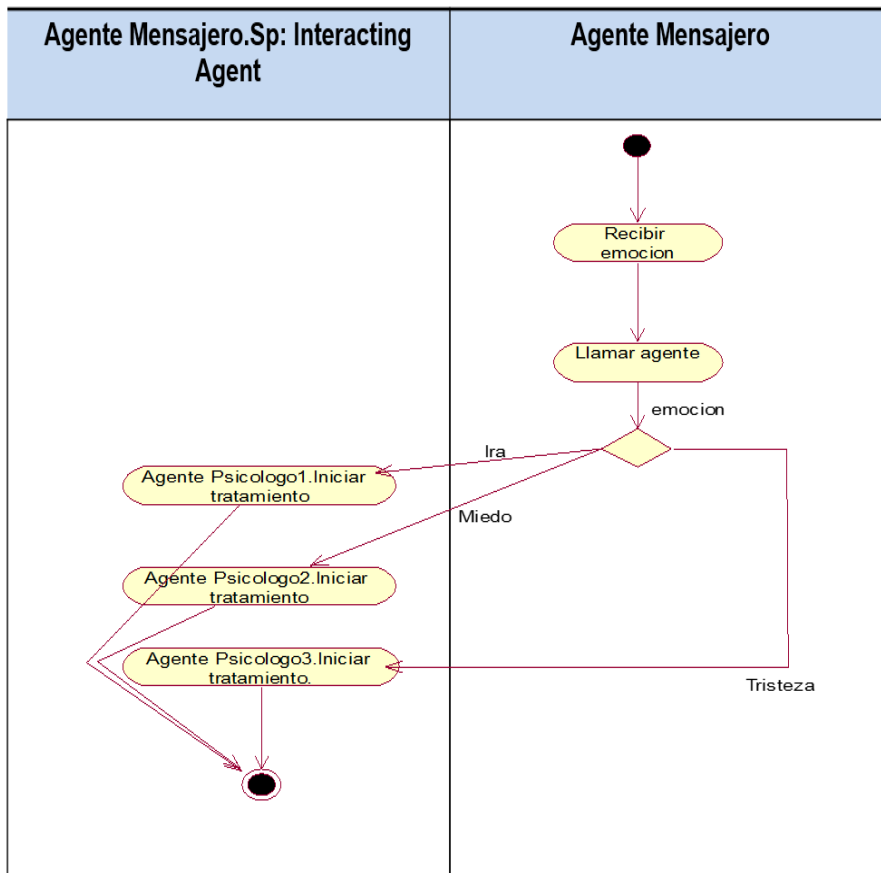


Figura 8: Diagrama de Actividades: Agente Mensajero

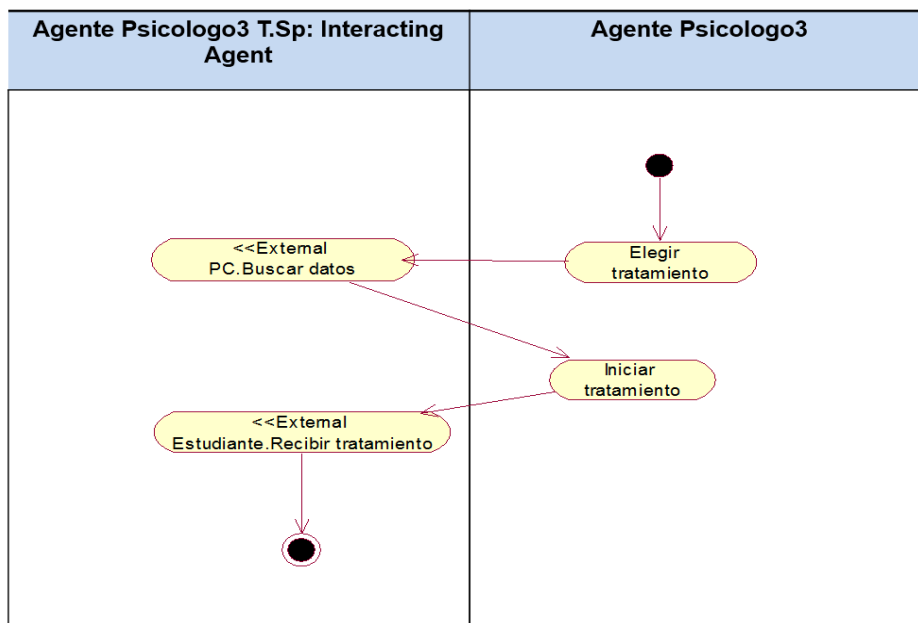


Figura 9: Diagrama de Actividades: Agente Psicologo3

2.2-Modelo de Sociedades de Agentes

Este nivel modela las interacciones sociales y dependencias entre los agentes involucrados en la solución. Las fases de este nivel son las siguientes:

2.2.1-Descripción de Ontología

Esta fase describe el conocimiento atribuido a agentes individuales y a sus interacciones mediante el uso de Diagramas de Clase.(PASSI 2003)

Esta etapa se divide en dos diagramas: descripción de ontología de dominio y descripción de ontología de comunicación.

2.2.1.1-Diagrama de Descripción de Ontología de Dominio

Del nivel de descripción obtenido en el Modelo de Requerimientos del Sistema, se obtiene una serie de clases clasificadas en términos de conceptos, acciones y predicados, que representan todos los diversos conceptos con que interactúan los agentes identificados .(PTK 2000)El diagrama de descripción de ontología de dominio es el encargado de detallar las clases anteriormente mencionadas.(Figura 10)

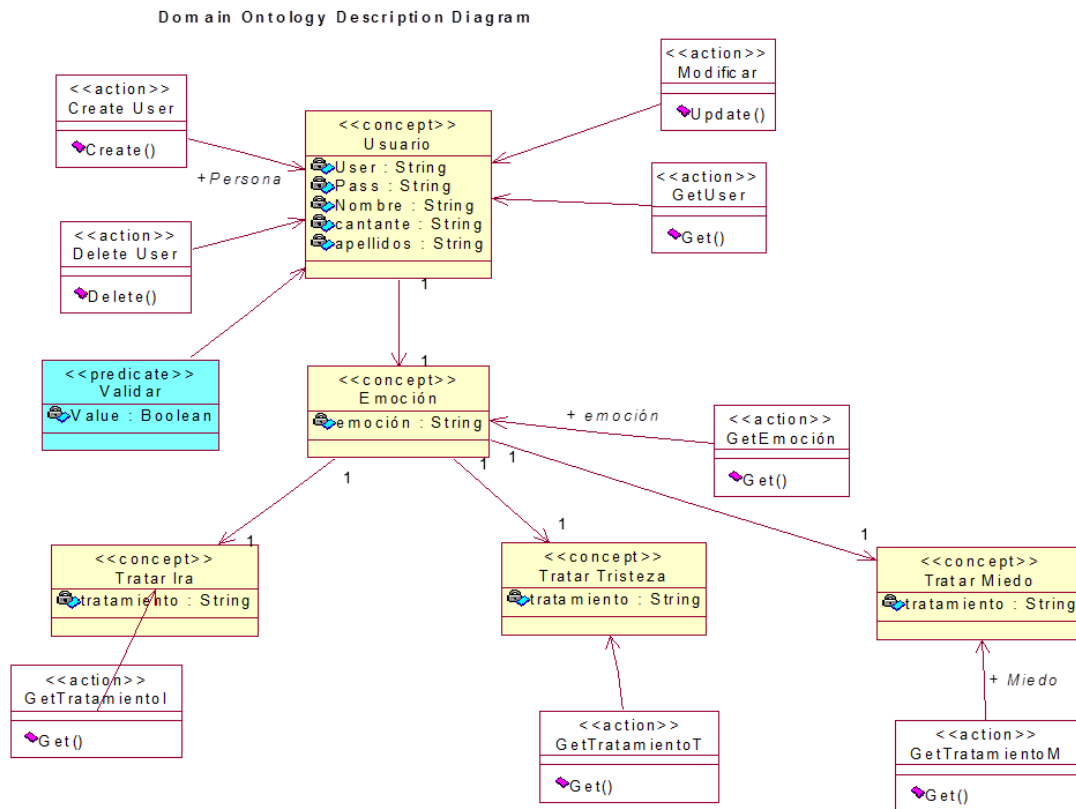


Figura 10: Diagrama de Descripción de Ontología del Dominio

2.2.1.2-Diagrama de Descripción de Ontología de Comunicación

Para cada comunicación es imprescindible especificar tres elementos: ontología, lenguaje y protocolo de interacción. Los lenguajes y protocolos de interacción son (en mayoría) estandarizados por FIPA. El encargado de esta tarea será Diagrama de Descripción de Ontología de Comunicación. (Figura 11)

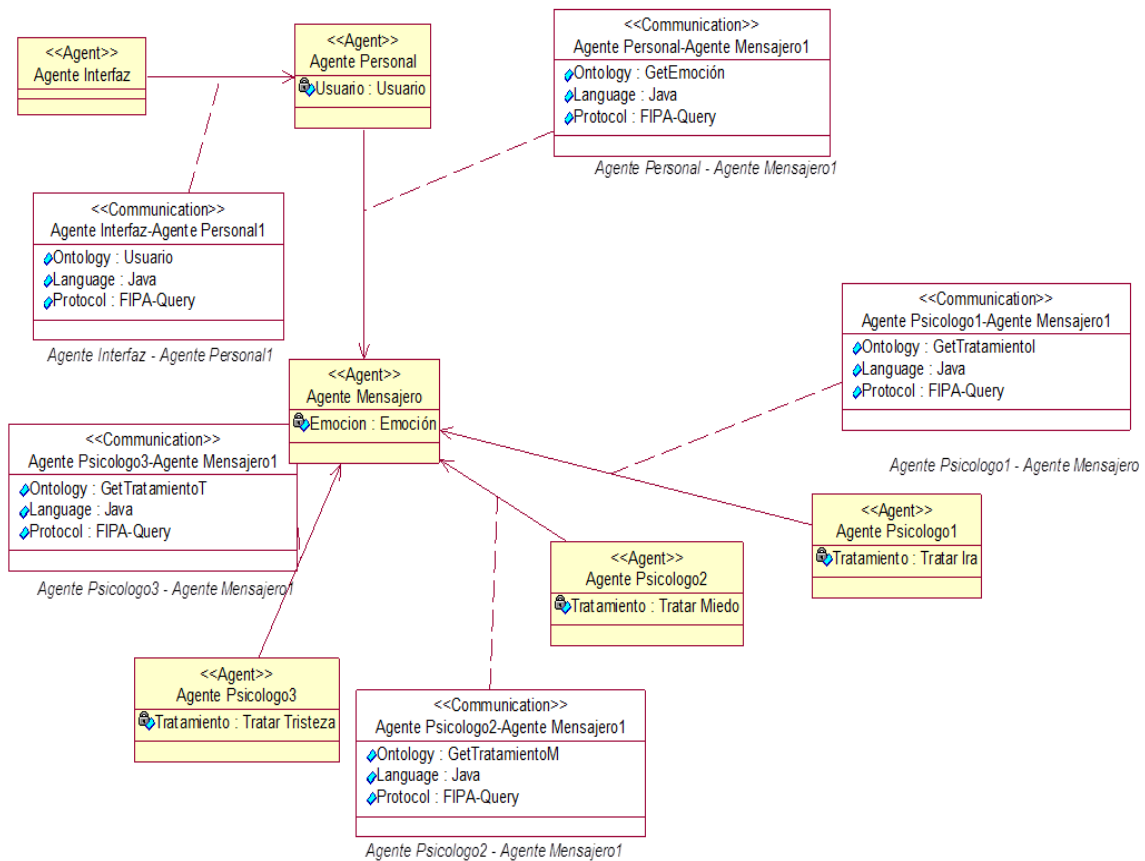


Figura 11: Diagrama de Descripción de Ontología de Comunicación.

2.2.2-Descripción de Roles

En esta etapa se introducen los agentes como clases y se especifican las tareas de cada uno dentro de los roles definidos anteriormente, estableciéndose también las comunicaciones entre cada rol.(Figura 12)

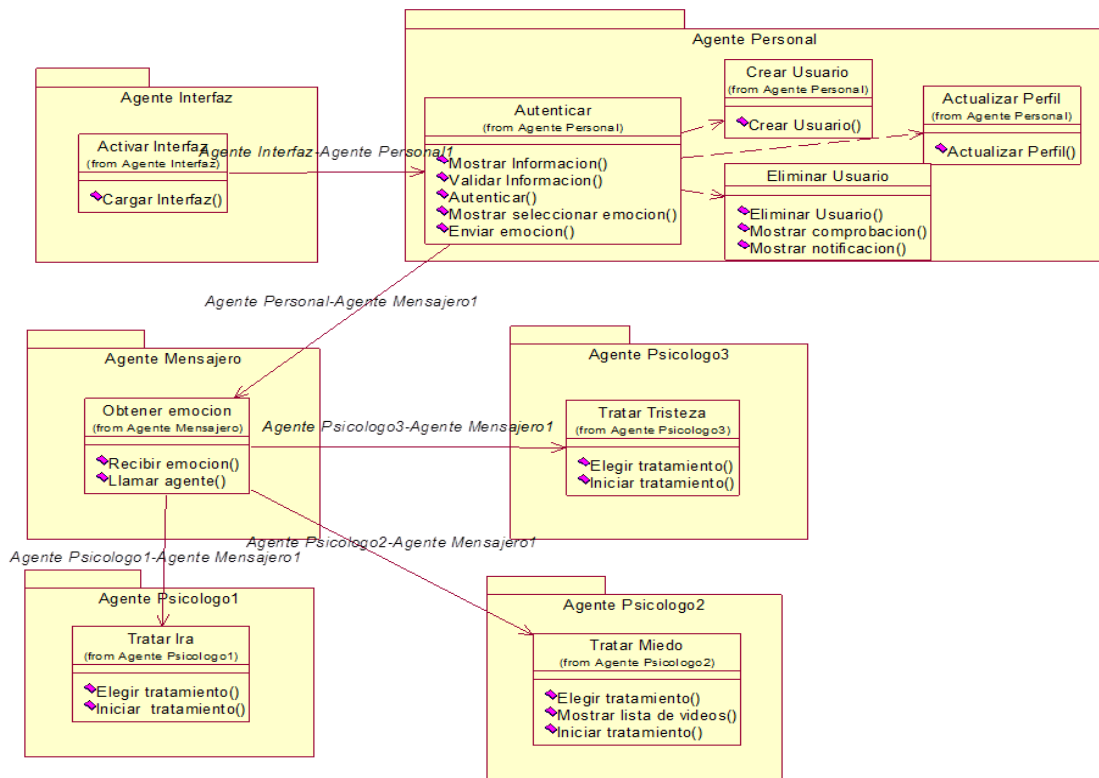


Figura 12: Diagrama de Descripción de Roles.

2.2.3-Descripción de Protocolos

Esta fase especifica la gramática de cada protocolo de comunicación en términos de “actos del habla” mediante el uso de Diagramas de Secuencia.(Figura 13)

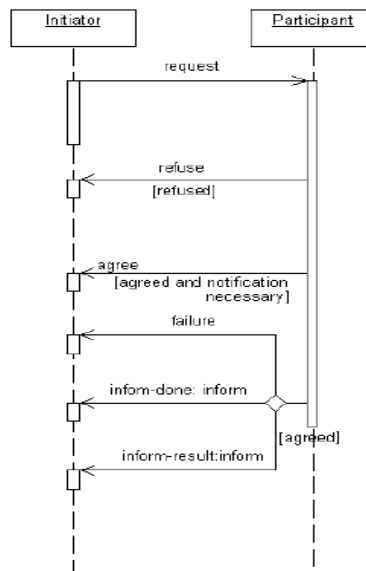


Figura 13: Diagrama de Descripción de Protocolos.

Conclusiones

Durante este capítulo se llevó a cabo el diseño del SMA utilizando PASSI, específicamente se desarrollaron las dos primeras fases, las cuales permitieron definir el entorno del sistema el cual facilitó en conjunto con el dominio, la confección por parte del programador, de los diferentes elementos que intervienen en el problema.

La ejecución de estas fases demostró que el sistema que se estaba construyendo era un sistema complejo y que una atención especial al detalle era imprescindible para el éxito de la comunicación entre agentes y del SMA en general.

Capítulo 3: Modelos de Implementación de Agentes

Introducción

El diseño del sistema multiagente así como las especificaciones de cada agente forma parte de la tercera fase del diseño utilizando PASSI, lo cual tiene gran importancia ya que se definen las funcionalidades y relaciones de los agentes, posibilitando una correcta implementación tras haber establecido una base sólida. Es por ello que en este capítulo se abordaran las fases de Implementación de agentes, el uso de código y el modelo de despliegue.

3.1-Modelo de Implementación de agente

En esta fase se describe la estructura de las clases de agente mediante dos diagramas: diagrama de estructura del SMA y diagrama de estructura del agente simple. Los cuáles serán los descritos a continuación.

3.1.1-Definición de estructura del SMA

Este diagrama presenta al SMA como un conjunto de clases, donde cada agente es representado como una clase y sus relaciones.(Figura 14)

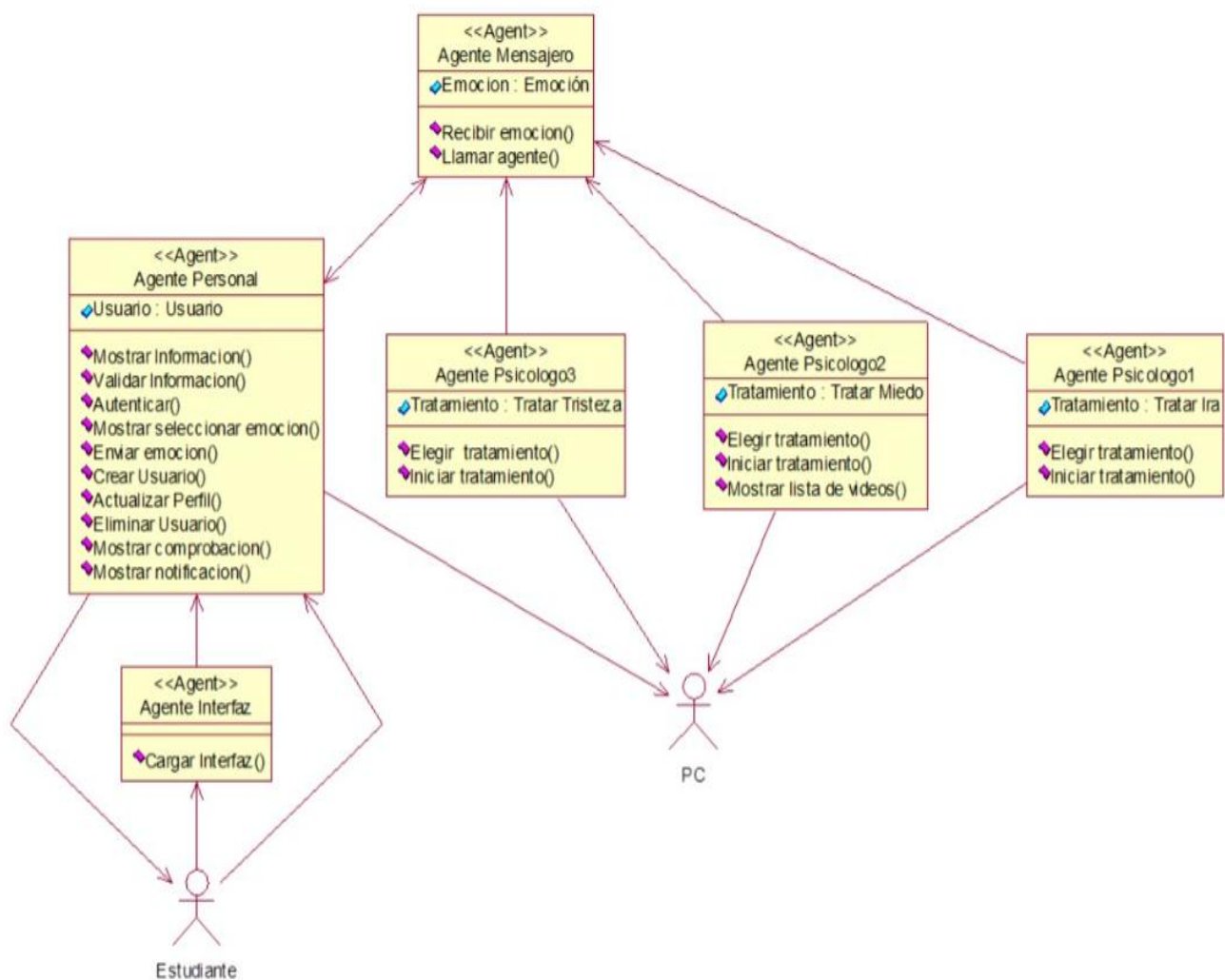


Figura 14: Diagrama Definición de Estructura del SMA.

3.1.2-Definición de estructura del agente simple

En este diagrama se describe la estructura del agente, sus funciones, así como la información que contendrá cada agente para su ejecución. En este caso los agentes a mostrar son: Agente Mensajero y Agente Psicologo3. Figura 15 y 16 (El resto ver Anexos 5 al 9)

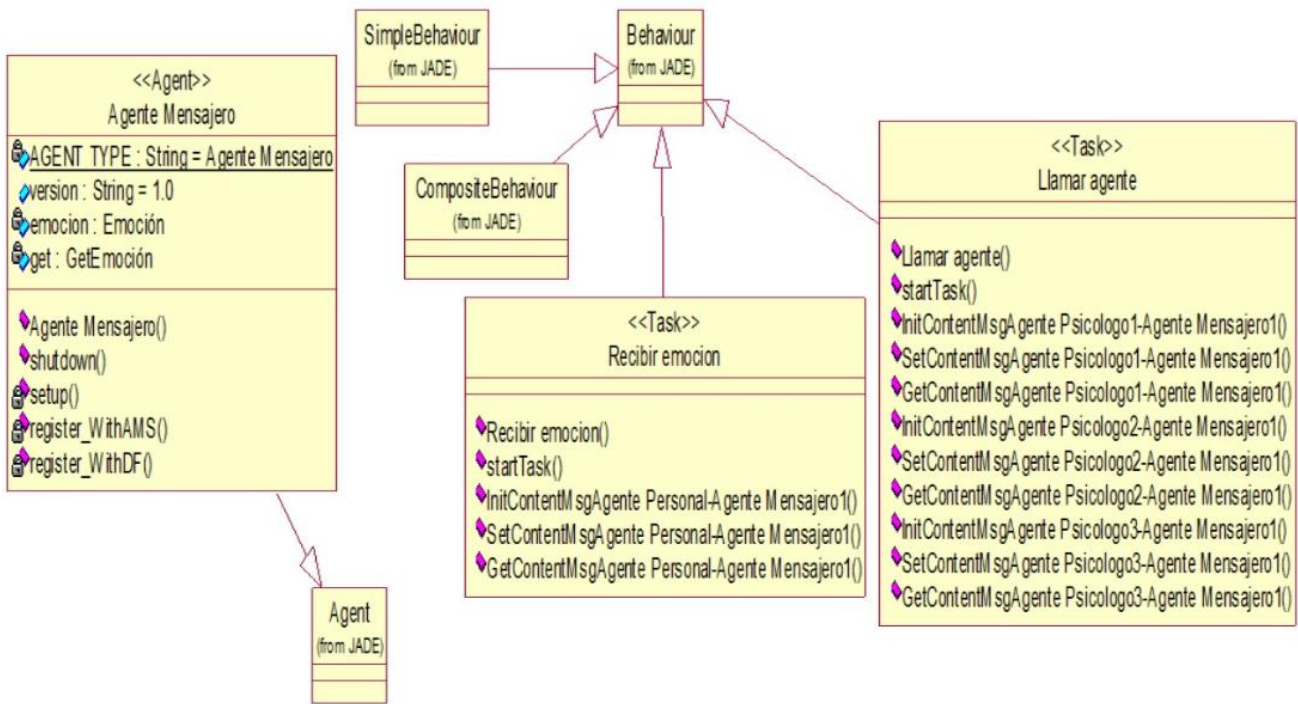


Figura 15: Diagrama Definición de Estructura del Agente Mensajero.

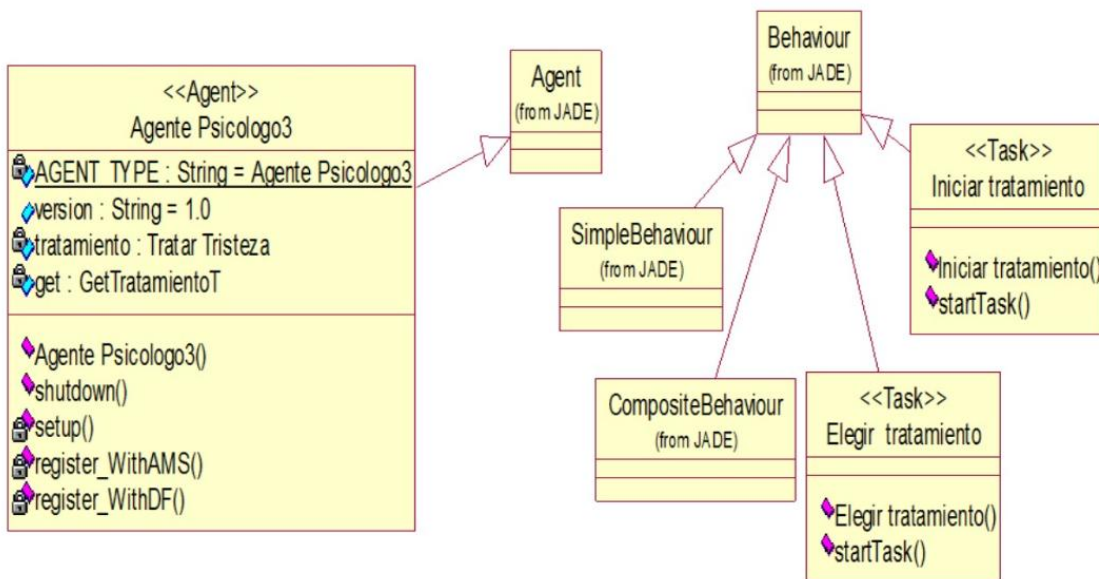


Figura 16: Diagrama Definición de Estructura del Agente Psicologo3

3.1.3-Descripción de conducta del agente

3.1.3.1-Descripción de conducta del SMA

En esta parte se muestra el flujo constante de información entre los agentes y de cada uno, mostrando esta descripción mediante diagramas de actividad, donde cada agente constituye una calle.

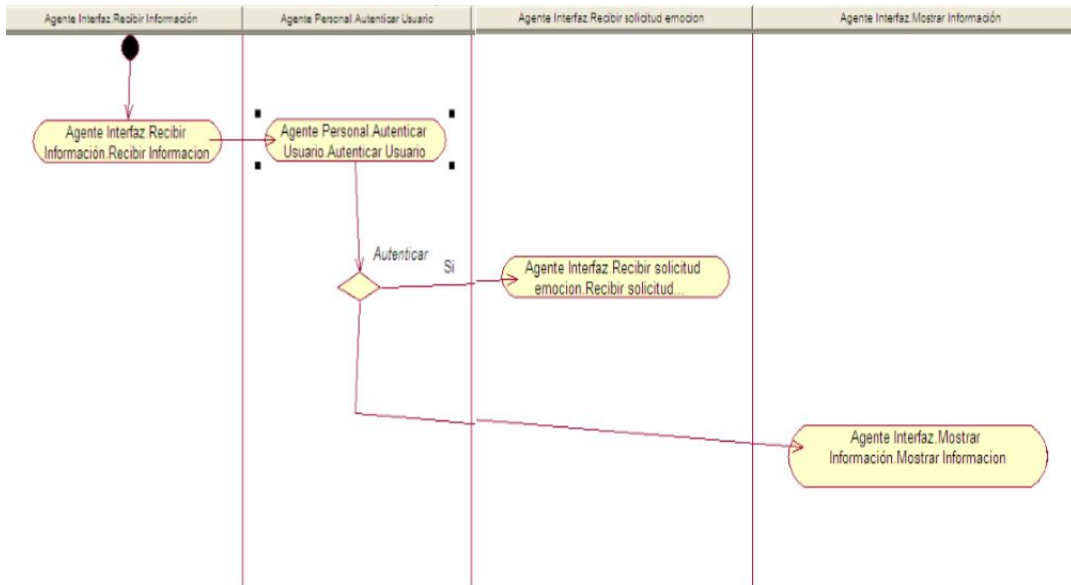


Figura 17: Descripción de conducta del SMA: Gestionar Usuario

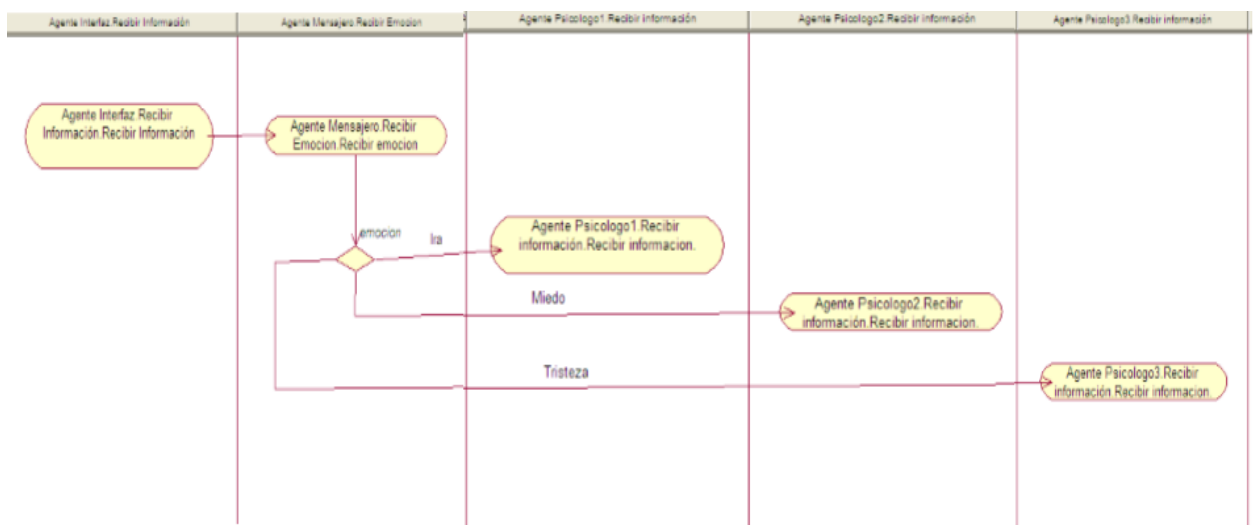


Figura 18: Descripción de conducta del SMA: Gestionar Emoción

3.1.3.2-Descripción de conducta del Agente simple

Está constituido por la implementación de los métodos introducidos en los Diagramas Definición de Estructura del Agente, la forma de escribirlo es libre, ya que es de acuerdo al criterio del diseñador. Esta fase no ha sido desarrollada pues no es objetivo de esta investigación. (Figura 19)

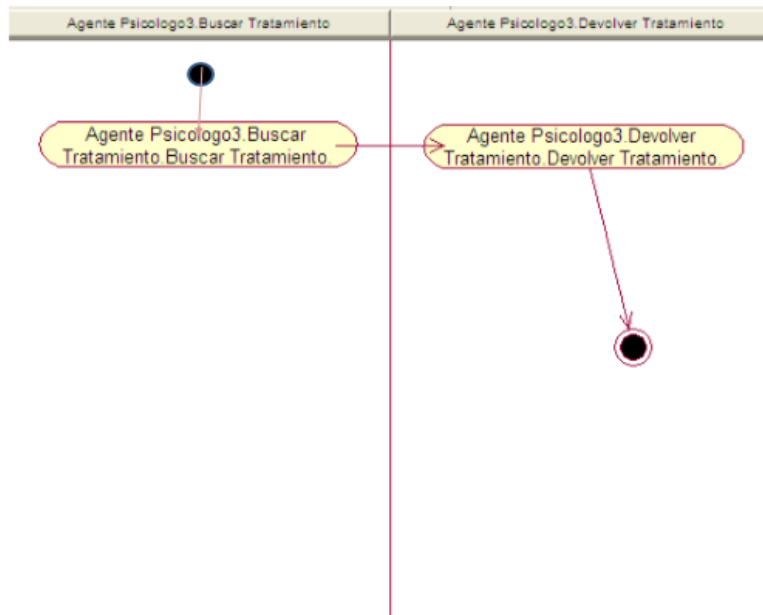


Figura 19: Descripción de Conducta del Agente Psicologo3.

3.2-Modelo de Código

3.2.1-Biblioteca de código reutilizable

El objetivo de esta fase es la implementación de las interfaces de cada agente, mediante la utilización de la herramienta PTK, pues esta genera el código base automáticamente, a partir del diseño de todas las fases anteriores. Como resultado de esto se obtuvo el siguiente código, el que en este caso solo se muestra el código generado del agente mensajero y el agente psicologo3.

Agente Mensajero

```
publicclass Agente Mensajero extendsAgent {  
  
private static String AGENT_TYPE = AgenteMensajero;
```

```
publicStringversion = 1.0;

private Usuario usuario ;

private Emoción emocion ;

privateModificarupDate ;

privateGetUsergetUser ;

private Create User createUser ;

privateDeleteUserdeleteUser ;

privateGetTratamientogetTratamiento ;

privateGetEmocióngetEmocion ;

private Tratar Ira tratarIra ;

private Tratar Tristeza tratarTristeza ;

private Tratar Miedo tratarMiedo ;

private Validar validar ;

public void AgenteMensajero(String platform, String name, String ownership) {

    //insert here your code

}

public void shutdown() {

    //insert here your code

}

private void setup() {
```

```
//insertthereyourcode

}

private void register_WithAMS() {

    //insert here your code

}

private void register_WithDF() {

    //insert here your code

}
```

Agente Psicologo3

```
public class Agente Psicologo3 extends Agent {

private static String AGENT_TYPE = Agente Psicologo3;

publicStringversion = 1.0;

private Emoción emocion ;

private Tratar Tristeza tristeza ;

privateGetTratamientogetTratamiento ;

privateGetEmocióngetEmocion ;

privateGetTratamientoTgetTratamientoT ;

public void Agente Psicologo3(String platform, String name, String ownership) {

    //insert here your code

}

public void shutdown() {
```



```

//insert here your code

}

private void setup() {

//insertthereyourcode

}

private void register_WithAMS() {

//insert here your code

}

private void register_WithDF() {

//insert here your code

}

```

3.3-Modelo de Despliegue

En este diagrama se establece la ubicación de cada agente, en todas las unidades de procesamiento.

3.3.1-Configuración de Despliegue

En esta parte se detalla la posición de los agentes en las unidades físicas presentes

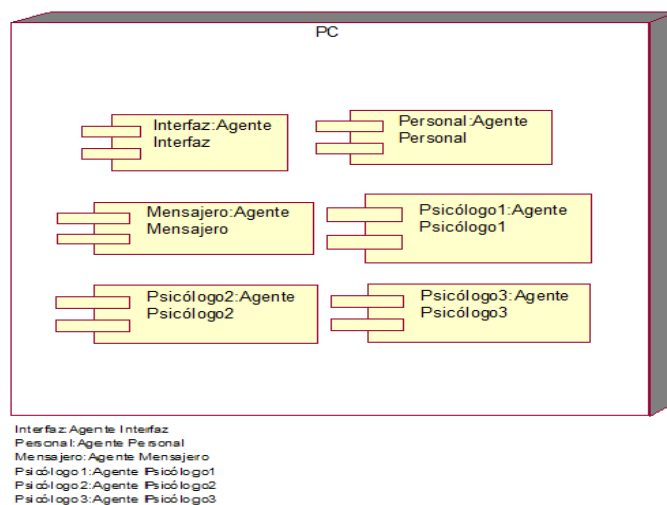


Figura 20: Diagrama de Despliegue.

3.4-Interfaces del SMA

Como resultado de la utilización de PASSI para guiar el desarrollo del SMA y de la integración de JADE y Java como IDE y lenguaje de programación surgieron las interfaces del sistema

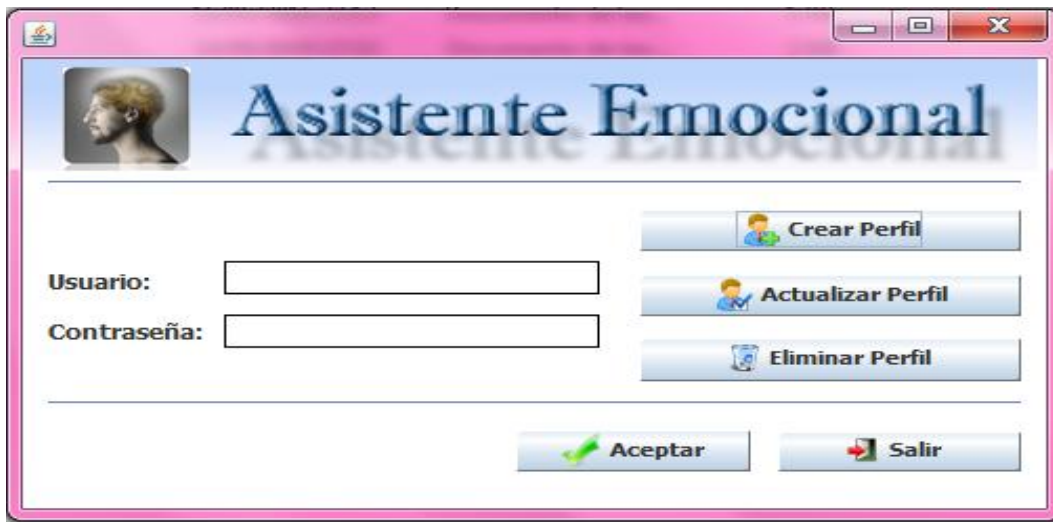


Figura 21: Interfaz: Gestionar usuario.

Esta primera interfaz será la encargada de gestionar los datos concernientes al usuario, ofreciéndole cuatro opciones: Acceder al sistema, Crear usuario, Actualizar Perfil y Eliminar usuario.



Figura 22: Interfaz: Obtener emoción.

Esta interfaz será la encargada de obtener la emoción seleccionada por el usuario.



Figura 23: Tratamientos de Tristeza: Fotos.



Figura 24: Tratamientos Tristeza: Música.

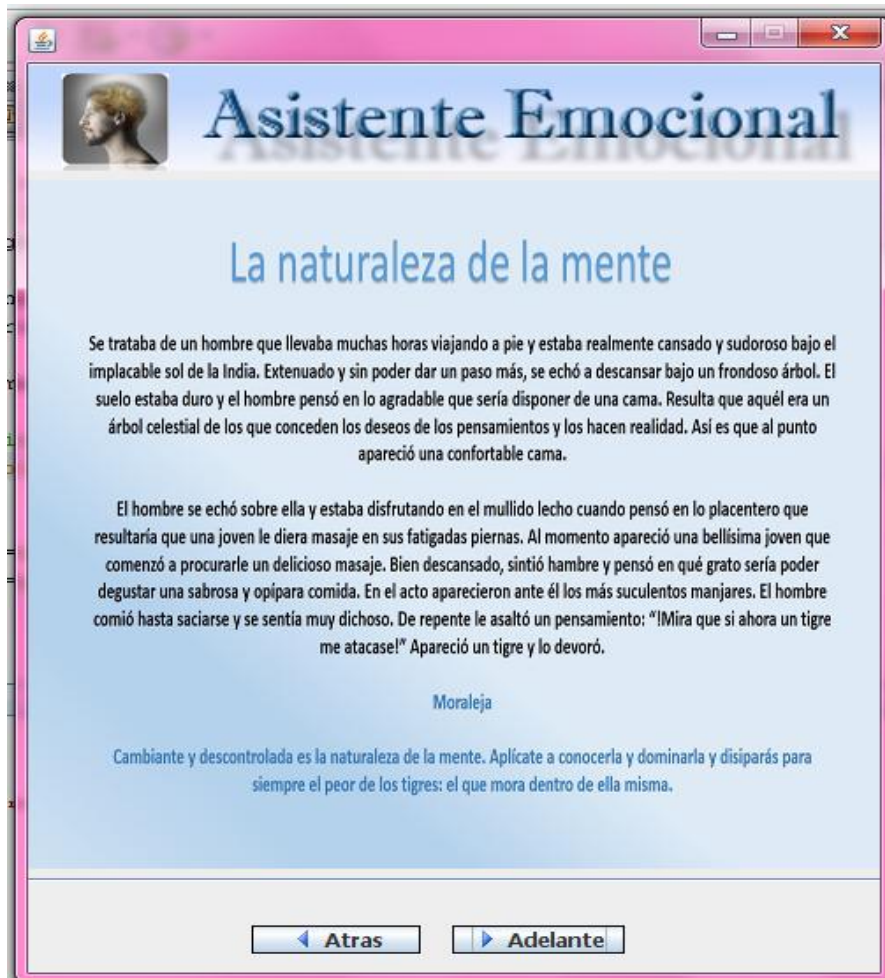


Figura 25: Tratamientos Tristeza: Fábulas



Figura 26: Tratamientos Tristeza: Frases

Estos serán los tratamientos que ejecutará el agente psicologo3, encargado de revertir la tristeza.

Conclusiones

El trabajo realizado durante este capítulo permitió establecer entre otras cosas, en términos de nodos la distribución de los agentes. Además demostró que JADE a pesar de utilizar al Netbeans como herramienta para su desarrollo, solo lo utiliza como cuerpo pues las clases a utilizar y la terminología a emplear son las propias de JADE y no las que ofrece el Netbeans.

Conclusiones

1. El estudio de los fundamentos teóricos metodológicos, permitió reconocer un campo poco explorado hasta ahora de la Inteligencia Artificial, denominado Agentes, así como permitió conocer la amplitud de la programación afectiva y la importancia en el manejo de las emociones utilizando herramientas computacionales sobre todo en la enseñanza.
2. Se pudo concluir que de entre todas las metodologías utilizadas, la más pertinente es Passi, ya que toma en cuenta todas las fases del ciclo de vida del desarrollo del software, aunque no existe una metodología estandarizada debido a la novedad del tema.
3. Se indicó que JADE a pesar de utilizar al Netbeans como herramienta para su desarrollo, es completamente independiente de este, pues posee sus propias clases y librerías, utilizando el Netbeans como instrumento para implementar sus métodos.

Recomendaciones

1. Establecer un orden de prioridad de acuerdo a los tratamientos a efectuar.
2. Debido al auge que existe actualmente con los dispositivos móviles, llevar este producto a una versión androide.

Bibliografía

Aguiar, E. (diciembre,2006). SQLite.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology.

Caire G., L. F., Evans R. (2001). Agent Oriented Analysis using MESSAGE/UML.

Canon-Videl, A. and J. J. Miguel-Tobal (2001). Emociones y Salud.Ansiedad y Estrés.

Cossentino, M. (2005). From requirements to code with the passi methodology. Agent oriented methodologies.

Domínguez-Dorado, M. (2006). NetBeans IDE 4.1. La alternativa a Eclipse.

FIPA (2002). "Foundation for Intelligent Physical Agents."

Huhns, M., Singh, M. P. (1998). Readings in Agents. Readings in Agents.

Javier García de Jalón , J. I. R. (2000). Aprenda Java como si estuviera en primero. UNIVERSIDAD DE NAVARRA.

JOSÉ PIQUERAS, V. R., AGUSTÍN MARTÍNEZ, LUIS OBLITAS (2010) EMOCIONES NEGATIVAS Y SALUD.

Mancilla, F. (2013). Proyecto Emociones Software Valparaíso, Universidad de Valparaíso.

Montero, F. (2006). "Language SQL."

Nwana, H. S. (1996). Software Agents: An Overview. Intelligent Systems Research. Ipswich, United Kingdom

PASSI (2003). "PASSI: a Process for Agents Societies Specification and Implementation."

PEKRUN, R. (1992). The Impact of Emotions on Learning and Achievement: Towards a Theory of Cognitive/Motivational Mediators.

Pinto, R. P. (junio,2006). Evaluacion de la Metodología PASSI en un Caso de Estudio, CUJAE.

PTK, T. (2000). Tutorial de la herramienta PASSI Tool-Kit.

R., L. R. (2008). Enseñando Prolog con mapas conceptuales. Revista Cubana de Ciencias Informática 2.

Rodríguez Elias, D. O. M. (mayo,2005) Desarrollo de Sistemas Multiagentes.

Rojas, L. M. H. (2014-2015). Diseño de un Sistema Multiagente para revertir las emociones negativas presentes en los estudiantes durante su interacción con un ambiente de enseñanza-aprendizaje virtual. Sancti Spiritus, Universidad "José Martí": 61.

Russell, S. (1996). Inteligencia Artificial: un enfoque moderno.

Stefan Poslad, P. B., Rob Hadingham (2005) The FIPA-OS agent platform: Open Source for Open Standards.

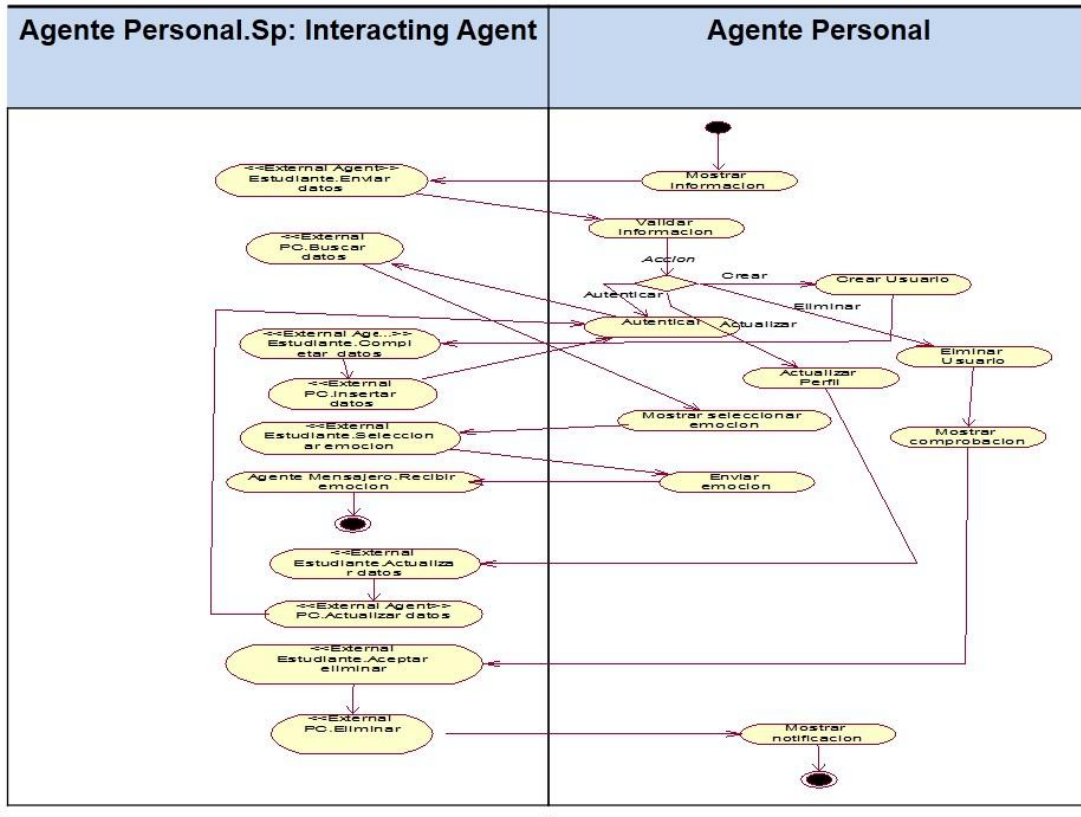
Tveit, A. (Mayo 2001) A Survey of Agent-Oriented Software Engineering.

V. Julian, V. B. (may-jun. 2000). Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial. NOVATICA.

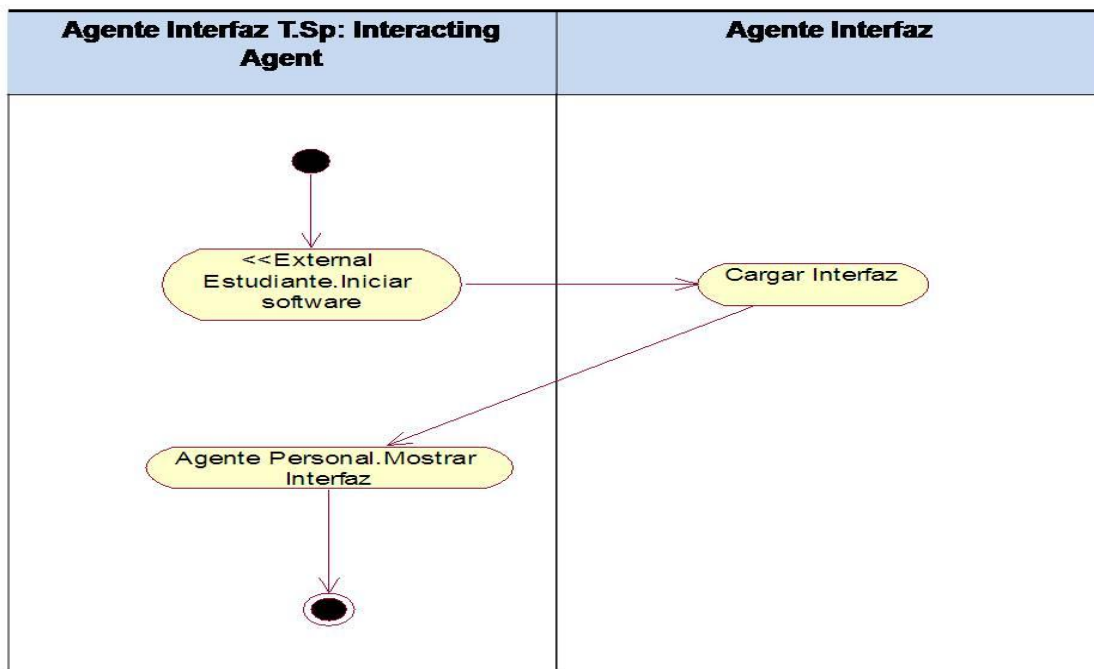
Whybrow, P. C., Akiskal, H.S. y McKinney, W.T. (1984). Mood disorder: Toward a new psychobiology. New York.

Zambonelli, F., Jennings, N. R., and Wooldridge, M. (2005). Multi-agent systems as computational organizations: the gaia methodology.

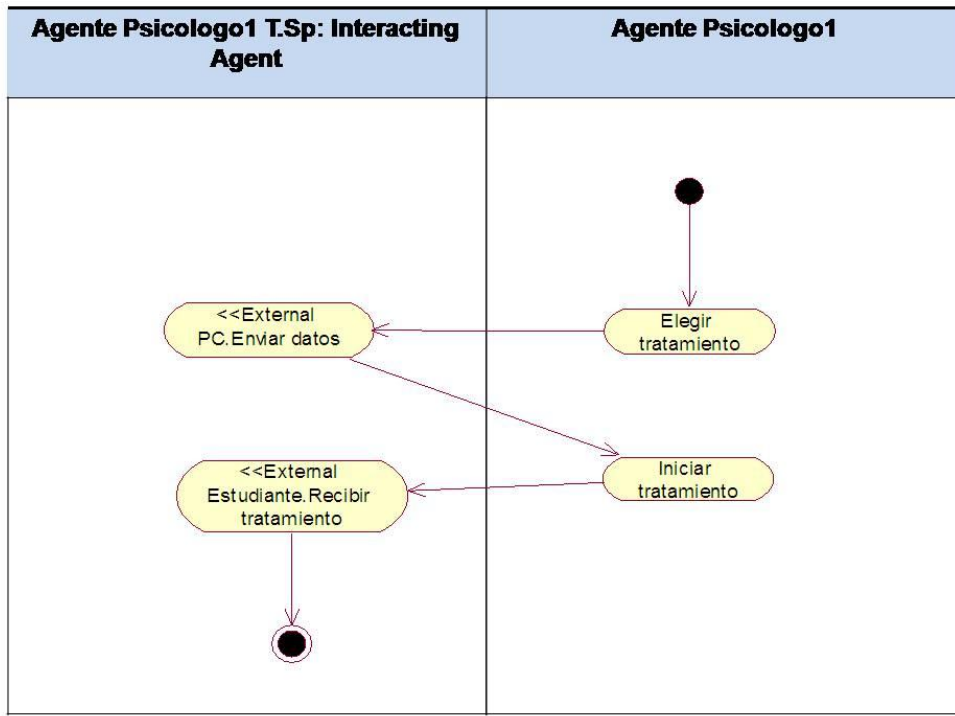
Anexos



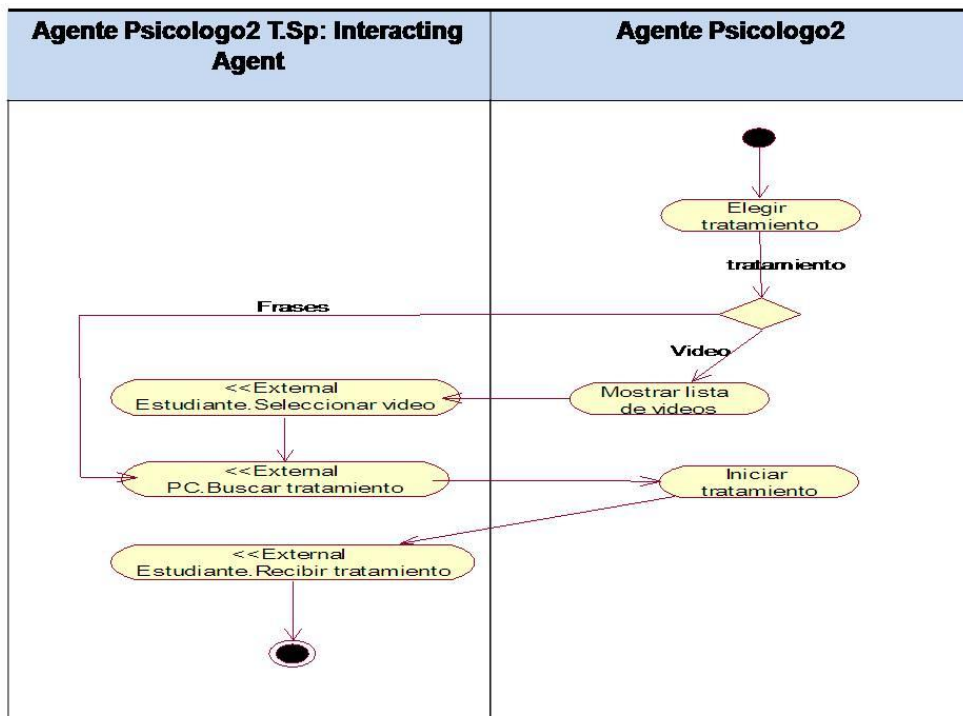
Anexo 1



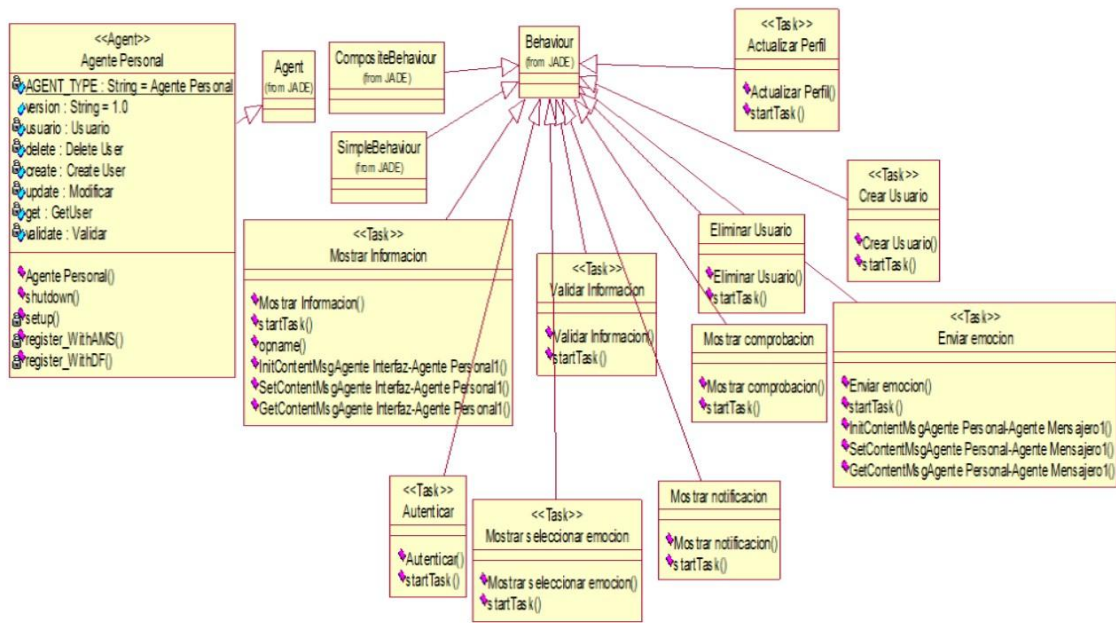
Anexo 2



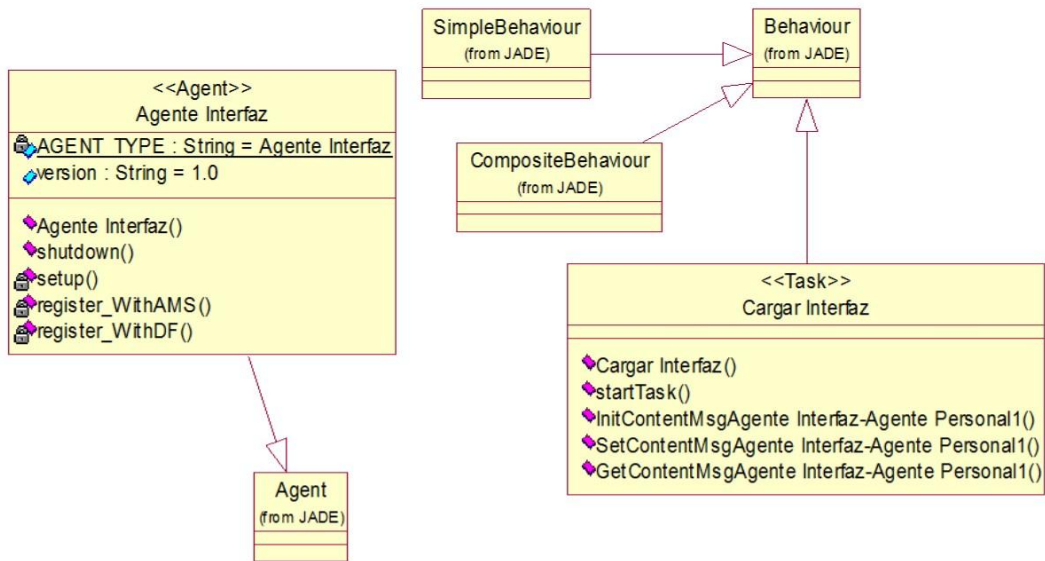
Anexo 3



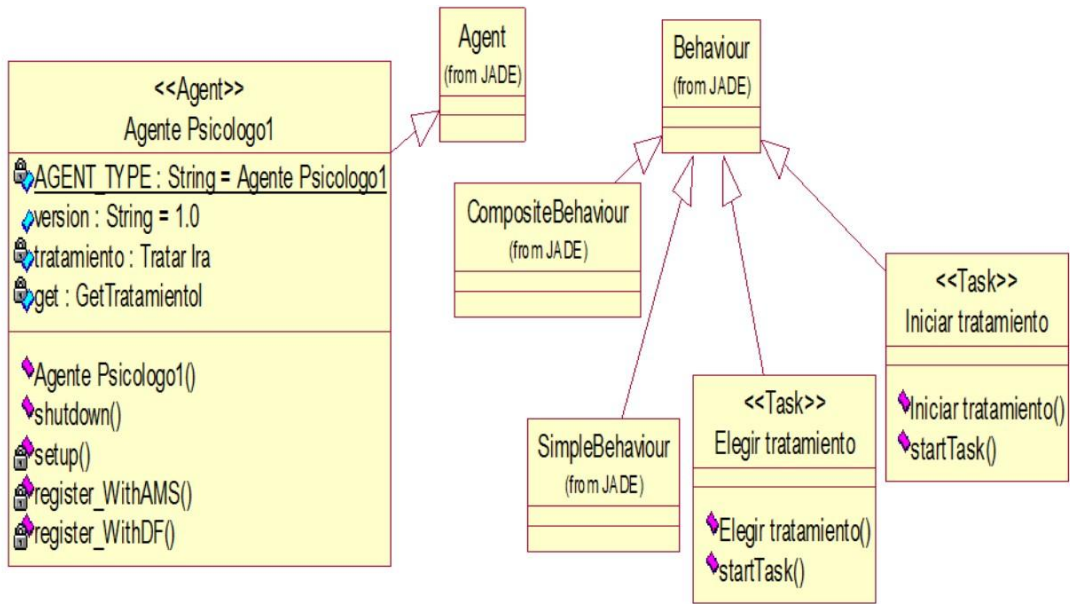
Anexo 4



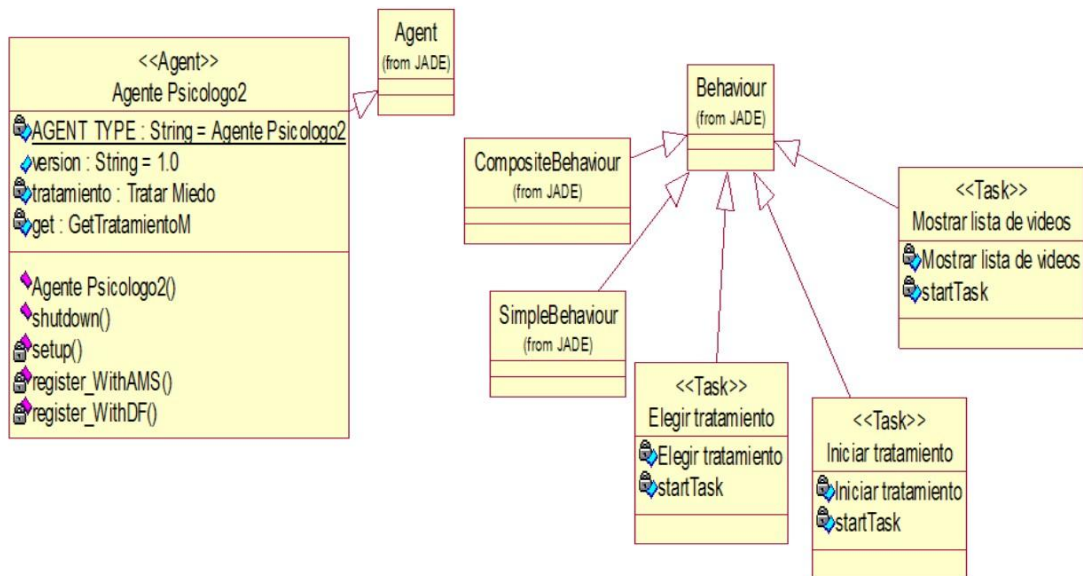
Anexo 5



Anexo 6



Anexo 7



Anexo 8