

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 ira presente en los estudiantes durante su interacción con un ambiente de enseñanza aprendizaje virtual

Autora: Maday Bermúdez Marrero

Tutora: MSc. Irina Machado Mutis.

Sancti Spíritus, Cuba

Curso 2015- 2016

Pensamiento.

*Los necios dan rienda suelta a su enojo, pero los sabios
calladamente lo controlan.*

Proverbios 29:11.

Agradecimientos:

A Dios, que es mi mayor tesoro, mi fuente de vida y la luz que ilumina mis pasos.

A mis padres, que son mi puerto seguro.

A mi hermano por llenar mi vida de alegría

A mi esposo, por ser mi oasis de paz y amor en cualquier tormenta.

A mis queridos profesores y amigos Firna, Lidia y Arley por ayudarme tanto, aun cuando el cansancio era inmenso.

A mis amigos Lisandra y Bonet; por estar juntos ayudándonos en todo momento de este proyecto.

A mis compañeros de aula por compartir conmigo estos 5 años, y enseñarme tantas cosas buenas de la vida.

A mis queridos Raúl, Madirmir, Liset, Oscar y Lisandra, por ser ustedes y estar para mí en tantos momentos que los necesité.

Contenido

Introducción:	10
Capítulo 1: SMA. Metodologías y herramientas.....	16
Introducción.....	16
1.1- Agentes.....	16
1.2- Sistemas Multiagente.....	17
1.3- Metodologías, herramientas y lenguajes a considerar para el desarrollo del sistema.....	18
1.3.1- Metodologías orientadas al diseño de SMA.	18
1.3.2- Plataformas de implementación orientadas a agentes.	20
1.4 Lenguaje de programación.....	22
1.5- Entorno de desarrollo integrado (IDE).....	22
1.6- Gestor de Base de datos.....	23
1.7- Emoción a revertir: La Ira.	23
1.7.1- Acercamiento a la Ira.	24
1.7.2- Tratamiento psicológico para la ira.	24
Conclusiones:.....	25
Capítulo 2: Diseño del SMA	27
Introducción:.....	27
2.2- Nivel 1: Modelo de requerimiento del sistema:.....	27
2.2.1- Descripción del entorno.....	27
2.2.2- Descripción del Dominio:	28
2.2.3- Identificación de agentes.	31
2.2.4- Identificación de Roles:.....	34
2.2.5- Especificación de Tareas.....	39
2.3- Nivel 2: Modelo de Sociedades de Agentes:.....	40
2.3.1- Descripción de la Ontología.	40
2.3.1.1- Descripción de ontología del Dominio.	40
2.3.1.2- Descripción de Ontología de Comunicación.	41
2.3.2- Descripción de Roles.	42
2.3.3- Descripción del Protocolo.....	43
Conclusiones:.....	44
Capítulo 3: Implementación de la solución propuesta.....	45
Introducción:.....	45
3.1- Nivel 3: Modelo de Implementación del Agente.....	45

3.1.1- Definición de estructura del Agente.....	45
3.1.2- Definición de Estructura del SMA:.....	45
3.1.2- Definición de estructura del agente simple.....	45
3.1.3- Descripción de Conducta del Agente.....	46
3.1.3.1- Descripción de Conducta del SMA.....	48
3.1.3.2- Descripción de conducta del agente simple.	49
3.2- Nivel 4: Modelo de Código.....	50
3.2.1- Biblioteca Reutilizable de Código.	50
3.2.2- Perfeccionamiento del código básico.	51
3.4- Nivel 5: Modelo de Despliegue.....	52
3.5- Interfaces del sistema:	52
Conclusiones.....	57
Conclusiones:.....	58
Recomendaciones:.....	59
Bibliografía:.....	60
Anexos.....	62

Índice de Figuras.

Figura 1 Descripción de agentes	17
Figura 2 Fases de PASSI.....	20
Figura 3. Diagrama de descripción del entorno.....	27
Figura 4: Diagrama de descripción del dominio.	32
Figura 5: Diagrama de identificación de agentes.....	34
Figura 6: Diagrama de secuencia del escenario: Gestionar Usuario.	35
Figura 7: Diagrama de secuencia del escenario: Gestionar Emoción.....	36
Figura 8: Diagrama de Actividad. Agente Personal.	39
Figura 9: Diagrama de actividad. Agente Psicólogo1.	40
Figura 10: Diagrama de Descripción de Ontología del Dominio.....	41
Figura 11: Diagrama de Descripción de Ontología de Comunicación.	42
Figura 12: Diagrama de Descripción de Roles.	43
Figura 13: Diagrama de Descripción de Protocolo.....	44
Figura 14: Diagrama de Descripción del SMA.....	46
Figura 15: Diagrama de Descripción del agente simple: Personal.....	47
Figura 16: Diagrama de Descripción Agente Simple: Psicólogo1.....	47
Figura 17: Diagrama de Descripción de conducta del SMA. Escenario gestionar usuario.....	48
Figura 18: Diagrama de Descripción de conducta del SMA. Escenario gestionar emoción.....	49
Figura 19: Diagrama de descripción del agente Psicólogo1.....	50
Figura 20: Diagrama de Despliegue.....	52
Figura 22: Ventana principal del SMA.	53
Figura 23: Ventana del SMA. Crear Usuario.....	54
Figura 24: Ventana del SMA. Eliminar Usuario.....	54
Figura 25: Ventana del SMA. Actualizar Usuario.	55
Figura 26: Ventana del SMA. Obtener Emoción.	55
Figura 27: Ventana del SMA: Tratamientos.	57

[Índice de tablas.](#)

Tabla 1 Descripción del CU. Cargar Interfaz.....	29
Tabla 2 Descripción del CU. Mostrar Información.....	29
Tabla 3 Descripción del CU. Recibir Información.....	29
Tabla 4: Descripción del CU. Validar Información.....	30
Tabla 5: Descripción del CU. Autenticar.	30
Tabla 6: Descripción del CU. Actualizar Perfil.....	30
Tabla 7: Descripción del CU. Crear Usuario.....	31
Tabla 8 Descripción del CU. Obtener Emoción.....	31
Tabla 9: Descripción del escenario: Gestionar Usuario.	37
Tabla 10: Descripción del escenario: Gestionar Emoción.....	38

Resumen

En la Universidad de Sancti Spiritus José Martí, en la actualidad se desarrollan investigaciones en el campo de la Inteligencia Artificial más específicamente en el desarrollo de Agentes, muchos son los logros alcanzados en este sentido; pero hasta el momento no se tienen en cuenta las emociones que presentan los estudiantes durante su interacción con estas herramientas. Es por eso que el objetivo de este trabajo es desarrollar un Sistema Multiagente (SMA) para revertir la ira como emoción negativa presente en los estudiantes universitarios durante su interacción con un ambiente de enseñanza aprendizaje virtual. Para ello se escogieron las siguientes herramientas y metodología, PASSI como metodología para el desarrollo de agentes, Jade como plataforma integrada al Netbeans para su implementación, Java y SQL como lenguajes de programación, y como gestor de base de datos SQLite. Teniendo en cuenta lo anterior se puede concluir que el estudio de los fundamentos teóricos metodológicos, demostró que un SMA era perfectamente capaz de resolver el problema al integrar diferentes mecanismos con el objetivo de poder aplicar correctamente los diferentes tratamientos establecidos por los psicólogos. Además las diferentes fases y niveles que plantea PASSI permitieron realizar la documentación del SMA y en especial, el modelo de código permitió la generación de código automático lo que facilitó el proceso de implementación. A pesar de que existen varias plataformas para desarrollar el SMA, Jade resultó ser el middle-ware adecuado para el desarrollo de esta investigación el cual al integrarse con netbeans como IDE permitió crear el Asistente Emocional, de manera tal que se pueda integrar a cualquier proyecto.

Abstract

In Sancti Spíritus University José Martí, nowadays, they develop investigations related to Artificial Intelligence, specifically Agent development, several achievement have been made in this sense; but so far they don't take into account the emotions the students present during their interaction with these tools. That is why the object of this works is to develop a Multiagent System to reverse the anger as negative emotion the student might prevent during their interaction with a virtual learning teaching environment. For I, the following tools were chooser, PASSI as agent development methodology, Jade as an integrated platform to netbeans for the implementation, Java and SQL as programing languages, and SQLite to manage the Data Base. Taking into account the above it can be concluded that the theoretical methodological foundations have proven that a MAS was perfectly capable of solving the problem integrating different mechanisms toging the right application of different treatments established by the psychologist. The different phases and levels in PASSU allowed the generation of automatic code making easier the process of implementation. Despite of the existence of several platforms to develop MAS, Jade turn out to be the right middle-ware to develop this investigation, which integrated with netbeans as IDE allowed to create Asistente Emocional, in a way that can be integrate to any project.

Introducción:

Los humanos son seres cambiantes, profundos, apasionados, muy parecidos; pero paradójicamente diferentes. En el centro de su ser se encuentran aquellas sensaciones que comúnmente son denominadas emociones, siendo estas: un estado complejo del organismo, generado habitualmente como respuesta a un acontecimiento externo o interno, caracterizado por una excitación o perturbación que predispone a una respuesta organizada (Alzina 2003). Las cuales pueden influir tanto positivamente, como negativamente en el individuo.

Dentro de la rama de las emociones, existen las llamadas emociones básicas que son aquellas que se caracterizan por una expresión facial característica y una disposición típica de afrontamiento. A pesar de que existe mucha polémica a la hora de decir cuáles son las emociones básicas, lo cierto es que las citadas con más frecuencia son: felicidad, tristeza, ira, sorpresa, miedo, disgusto y vergüenza. (Alzina 2003)

Dentro de este grupo de emociones básicas, se pueden encontrar emociones negativas y positivas. Estas últimas son agradables y se experimentan cuando se alcanza algún logro o meta, así como cuando algún acontecimiento externo provoca una reacción favorable, en el individuo. Hay divergencia en el pensamiento de muchos psicólogos al ponerse de acuerdo cuáles son las emociones positivas, pero las que más predominan son: alegría, orgullo, satisfacción, esperanza, fluidez, elevación (Jiménez 2006).

Aun cuando se han mencionado y definido cuáles son aquellas emociones positivas, el centro de atención no serán ellas, sino las que la psicología ha denominado como emociones negativas: El miedo-ansiedad, la ira, la tristeza-depresión y el asco son reacciones emocionales básicas que se caracterizan por una experiencia afectiva desagradable o negativa y una alta activación fisiológica (José A. Piqueras Rodríguez, Victoriano Ramos Linares et al. 2009).

Pero dentro de este grupo de emociones, la que más invadirá este espacio es la que es nombrada como ira, definida por muchos psicólogos como una emoción sumamente destructiva, hace referencia a sentimientos de violencia, enojo, angustia e indignación provocados por factores o personas externas al

individuo. Dentro de sus principales características podremos encontrar según (Chóliz 2005):

- ✓ Estimulación aversiva, tanto física o sensorial, como cognitiva.
- ✓ Condiciones que generan frustración, interrupción de una conducta motivada, situaciones injustas, o atentados contra valores morales.
- ✓ Inmovilidad, restricción física o psicológica.
- ✓ Elevada actividad neuronal y muscular.
- ✓ Reactividad cardiovascular intensa (elevación en los índices de frecuencia cardíaca, presión sistólica y diastólica).
- ✓ Focalización de la atención en los obstáculos externos que impiden la consecución del objetivo o son responsables de la frustración.
- ✓ Obnubilación, incapacidad o dificultad para la ejecución eficaz de procesos cognitivos.
- ✓ Movilización de energía para las reacciones de autodefensa o de ataque.
- ✓ Eliminación de los obstáculos que impiden la consecución de los objetivos deseados y generan frustración. Si bien la ira no siempre concluye en agresión, al menos sirve para inhibir las reacciones indeseables de otros sujetos e incluso evitar una situación de confrontación.
- ✓ Sensación de energía e impulsividad, necesidad de actuar de forma intensa e inmediata (física o verbalmente) para solucionar de forma activa la situación problemática.
- ✓ Se experimenta como una experiencia aversiva, desagradable e intensa. Relacionada con impaciencia.

En cuanto a su expresión; tres son sus diferentes estilos:

1. Supresión de la ira (anger-in). Afrontamiento de la situación reprimiendo la expresión verbal o física, aunque persista activación alta.
2. Expresión de la ira (anger-out). Afrontamiento de la situación manifestando conductas airadas verbal o físicamente, hacia otras personas u objetos, aunque sin intención de producir daño.
3. Control de la ira (anger-control). Afrontamiento de la situación canalizando la energía emocional proyectándola hacia fines constructivos para maximizar la posible resolución positiva del conflicto.

Dado las afectaciones que esta emoción y otras pueden provocar en los humanos, en el Massachusetts Institute of Technology (MIT) se desarrolla desde el año 1997 por Rosalind Picard un proyecto de investigación denominado Computación Afectiva que es una disciplina de la Inteligencia Artificial que intenta desarrollar métodos computacionales orientados a reconocer emociones humanas y generar sintéticas.

Utilizando la Computación Afectiva se han realizado a nivel mundial varias aplicaciones. Afectiva, una empresa incipiente surgida del Laboratorio de Medios del MIT, están trabajando en *software* que entrena computadoras para reconocer las emociones humanas con base en sus expresiones faciales y respuestas fisiológicas.

Una compañía llamada Beyond Verba está trabajando en una herramienta de *software* que puede analizar el habla y, con base en el tono de voz de una persona, determinar si indica cualidades como arrogancia o molestia, o ambas.

Como podemos ver varias entidades de renombre están tratando de desarrollar este tipo de computación, sin embargo, aun cuando estas aplicaciones de *software* son capaces de identificar emociones negativas no intentan revertirlas; aun así, es notable el avance que alcanza esta arista de la Inteligencia Artificial en algunas ramas de la sociedad; más en el contexto de la educación tiene una aplicación escasa.

De igual manera, dentro del horizonte cubano esta línea de investigación es muy poco transitada; no obstante en la Universidad de Sancti Spíritus José Martí Pérez (UNISS) se desarrollan investigaciones en este sentido, tal es el caso de APA-PROLOG que no es más que un Sistema Multiagente que toma en cuenta los conocimientos previos del estudiante, y contribuye al proceso de enseñanza-aprendizaje de los principales contenidos, de la programación lógica. Incluye el uso de mapas conceptuales. (Rodríguez 2008)

Así mismo (Sánchez, 2014) en su trabajo de diploma propone un sistema experto, que simula el trabajo de un psicólogo para determinar la emoción vigente en el diagnosticado.

De la misma manera (Rojas 2015) en su tesis de grado propone el diseño de un Sistema Multiagente utilizando la metodología PASSI, si bien en él se propone diseñar un Sistema Multiagente destinado a revertir la emoción negativa Ira, Miedo y Tristeza en los estudiantes, hasta el momento no se ha desarrollado la herramienta.

En este punto se puede decir que hasta donde llega el horizonte de información recopilado, varias son las herramientas que se desarrollan con el objetivo de que el estudiante pueda aprender, pero ninguna de ellas contempla las emociones negativas que estos puedan sentir y mucho menos intenta revertirlas.

Por estas razones se considera la necesidad de desarrollar un sistema capaz de revertir la Ira como emoción negativa.

De esta manera se puede especificar el **problema de investigación** siguiente:

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

Para dar solución a este problema se ha trazado como **objetivo general**: Desarrollar un Sistema Multiagente (SMA) para revertir la Ira como emoción negativa presente en los estudiantes universitarios durante su interacción con un ambiente de enseñanza aprendizaje virtual. Para cumplir con este objetivo se definieron las siguientes preguntas de investigación:

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

Para dar solución a estas preguntas de investigación se trazaron las siguientes tareas de investigación:

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

Este trabajo estará compuesto por una introducción, tres capítulos, conclusiones y recomendaciones y en él la investigación profundizará en el tema de la siguiente forma:

Capítulo 1:

En este capítulo se definirán los conceptos esenciales para el desarrollo del sistema, entre ellos se destacan Agentes y Sistema Multiagente (SMA). De la misma manera se realizará un estudio de las herramientas necesarias, entre ellas las mencionadas son las siguientes: Metodologías para el diseño de un SMA, plataformas para la implementación de un SMA, Netbeans, SQL, SQLite y entre los lenguajes de programación el desarrollado será Java.

Capítulo 2:

En este capítulo se desarrollará el diseño del SMA de acuerdo a la metodología seleccionada, PASSI, contemplando solamente sus dos primeros niveles, recorriendo cada una de sus fases

Capítulo 3:

En este espacio se desarrollará lo referente a la implementación del sistema, comenzando por el nivel 3 de PASSI, desarrollando cada una de las fases que

ellos encierran. En una de estas últimas, se generará una biblioteca de código reutilizable que se utiliza como punto de partida para la creación del SMA.

Capítulo 1: SMA. Metodologías y herramientas.

Introducción

Este capítulo propone un acercamiento a los principales conceptos asociados a la computación afectiva, tales como agentes y SMA, también se referirá a las diferentes plataformas de diseño e implementación que existen para el desarrollo de esta aplicación de software y los lenguajes utilizados para ello, de la misma manera recogerá qué es la ira, cuál es el tratamiento desde el punto de vista psicológico y cuál es el correspondiente desde la perspectiva de la computación afectiva.

1.1- Agentes.

Como consecuencia del progreso acelerado de la tecnología entra en escena la computación afectiva. Con ella se hace, cada vez, más real el hecho de que las computadoras puedan percibir algunas de las emociones humanas.

En este punto se nos presenta la siguiente interrogante: ¿Cómo lograr que una computadora tome en cuenta los estados afectivos de los estudiantes?

Según (Picard 1997) las características que serían necesarias para proveer a las computadoras con los medios para tomar en cuenta los estados afectivos de los estudiantes, serían: reconocer emociones, expresar emociones, tener emociones y por último tener inteligencia emocional.

Como es preciso cubrir el aspecto de la necesidad de que la computadora tenga inteligencia emocional para poder consumir la computación afectiva, es necesario recurrir al uso de los Agentes Inteligentes (AI) que de acuerdo con (Bergolla, 2009) son una entidad que actúa 'de manera autónoma' en un entorno, transformándolo mediante la interrelación con otras entidades, trabajando para alcanzar un fin determinado.

Sobre este criterio se ve a un agente como una entidad que a través de sus sensores descubre su entorno, y a través de sus percepciones decide qué acciones debe realizar sobre él para transformarlo. (Figura 1)

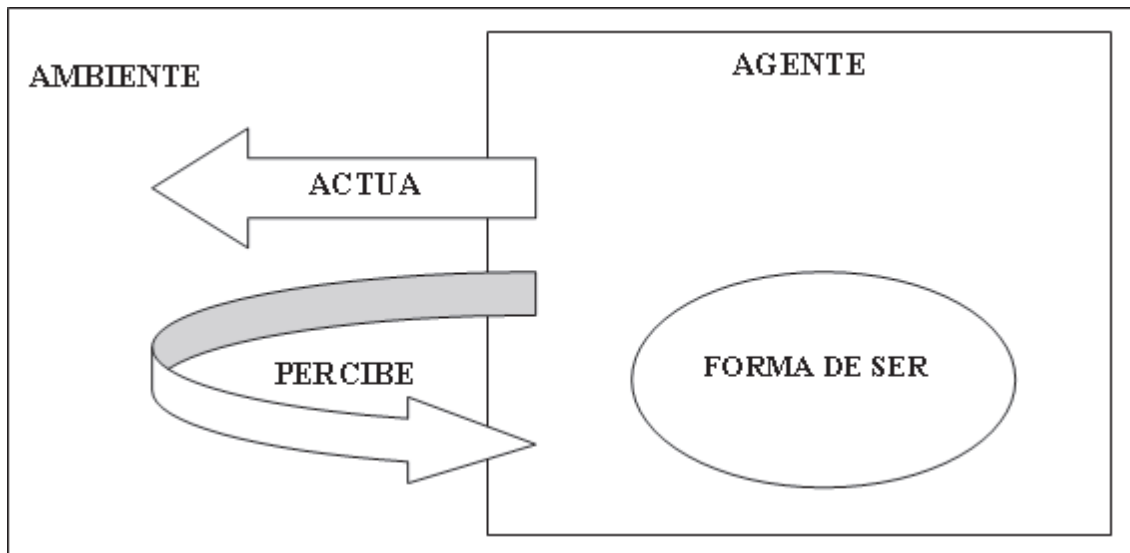


Figura 1 Descripción de agentes

Dentro de sus principales características se destacan (tomado parcialmente de (Pinto 2006)):

- **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.
- **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

1.2- Sistemas Multiagente.

A pesar de la gran cantidad de tareas que cada agente puede realizar, consiguen muy poco cada uno por sí mismo, de forma separada; por eso es importante la interrelación y conexión de manera óptima y segura entre ellos.

De esta importancia surgen los SMA que no son más que la interacción y colaboración de cada uno de los agentes que conforman el sistema.

Un concepto más elaborado sería: Grupo de agentes que interaccionan entre sí para lograr objetivos comunes, que a su vez está compuesto por un *entorno*, un

conjunto de objetos integrados ha dicho entorno (que a su vez pueden ser creados, modificados y/o destruidos por los agentes)(Pinto 2006).

Para que exista un correcto funcionamiento del SMA es imprescindible la comunicación eficaz entre los agentes que los componen, la cual se desarrolla a través de mensajes.

Entre las características más relevantes y necesarias de los SMA sobresalen: la organización social, cooperación, coordinación, negociación y control (Gómez 2009).

1.3- Metodologías, herramientas y lenguajes a considerar para el desarrollo del sistema.

Con el objetivo de hallar las metodologías, herramientas y lenguajes correctos para el desarrollo del SMA es necesario realizar un análisis detallado de alguna de las más utilizadas.

1.3.1- Metodologías orientadas al diseño de SMA.

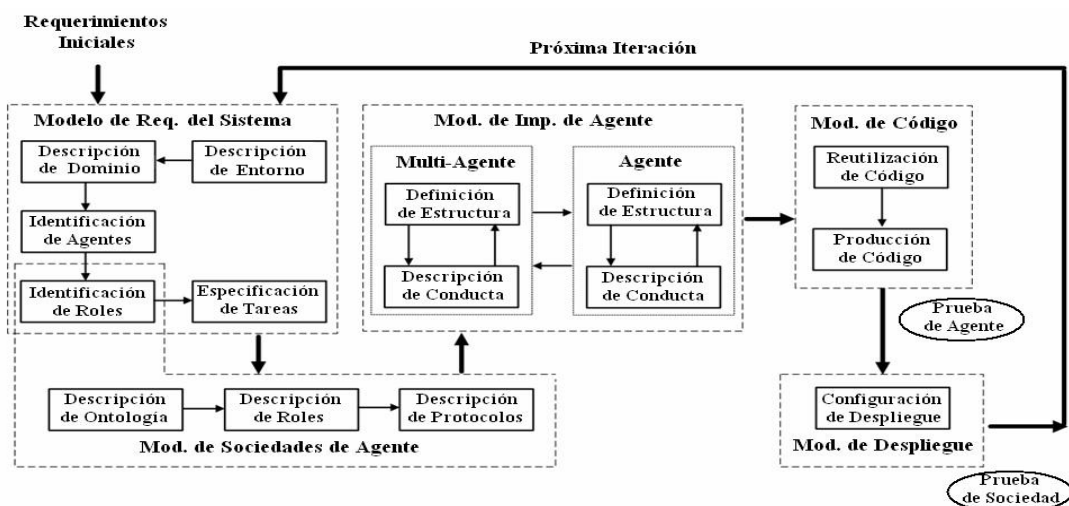
Los agentes no dejan de ser software y como tales demandan de análisis de requerimientos, especificaciones de diseño, pruebas,etc, si se apela a las metodologías tradicionales se perderá el tiempo, pues ellas no recogen las necesidades y exigencias que necesita este nuevo estilo de programación orientada a agentes. De aquí la necesidad de crear metodologías adecuadas que colaboren con la meta de diseñar un SMA viable.

Entre las metodologías más renombradas se encuentran:

- MAS-CommonKADS: La metodología CommonKADS gira alrededor del modelo de experiencia y está pensada para desarrollar sistemas expertos que interactúen con el usuario. De hecho considera sólo dos agentes básicos: el usuario y el sistema. MAS- CommonKADS extiende los modelos de CommonKADS para tener en cuenta la posibilidad de que dos o más componentes del sistema interactúen.

MAS-CommonKADS ha sido la primera en plantear un desarrollo de SMA integrado con un ciclo de vida de software, concretamente el espiral dirigido por riesgos. (Sanz 2003)

- BDI: Según esta teoría, los agentes utilizan un modelo del mundo, una representación de cómo se les muestra el entorno. El agente recibe estímulos a través de sensores ubicados en el mundo. (Sanz 2003)
- MaSE: Posee parte del paradigma orientado a objetos y asume que un agente es sólo una especialización de un objeto. La especialización consiste en que los agentes se coordinan unos con otros vía conversaciones y actúan proactivamente para alcanzar metas individuales y del sistema. (Sanz 2003)
- INGENIAS: Define un conjunto de meta-modelos (una descripción de alto nivel de qué elementos tiene un modelo) con los que hay que describir el sistema. Los meta-modelos indican qué hace falta para describir: agentes aislados, organizaciones de agentes, el entorno, interacciones entre agentes o roles, tareas y objetivos. Estos meta-modelos se construyen mediante un lenguaje de meta-modelado. (Sanz 2003)
- Passi: La metodología 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.

Figura2 Fases de PASSI

De todas las mencionadas anteriormente la seleccionada como más idónea para desarrollar un diseño certero es PASSI; pues ella incorpora cinco modelos que responden a diferentes niveles de diseño, los que a su vez se subdividen en doce fases que resultan en el desarrollo del ciclo de vida de un SMA. Los mecanismos que implementa permiten la personalización de la representación del diseño orientado a agentes, evitando con ello la adopción de un lenguaje de modelación totalmente nuevo. (Pinto 2006)

1.3.2- Plataformas de implementación orientadas a agentes.

Como es imprescindible una metodología para el diseño del SMA es innegable la necesidad de un marco de trabajo, el cual facilite la etapa de implementación. Tomando en cuenta cuáles de estas plataformas de implantación logran una correcta integración con PASSI se destacan las que siguen:

➤ JACK:

JACK provee un entorno de desarrollo orientado a agentes construido sobre Java y completamente integrado con este lenguaje de programación. Incluye todos los componentes del entorno de desarrollo de Java así como también ofrece extensiones específicas para implementar el comportamiento de los agentes

➤ JAFMAS (Java Framework for Multi-agent Systems):

JAFMAS provee una metodología genérica para desarrollar sistemas multiagente basados en los actos del habla junto con un conjunto de clases para soportar la implementación de estos agentes en Java.

➤ ZEUS:

El objetivo del proyecto ZEUS es facilitar el desarrollo rápido de nuevas aplicaciones multiagente mediante la abstracción de los principios y componentes más comunes a una herramienta.

La idea es proveer una herramienta de propósito general y personalizable, que permita la creación de agentes colaborativos y que pueda ser usada por

ingenieros de software con poca experiencia en tecnología de agentes para crear sistemas multiagente.

➤ JADE (Java Agent Development Framework):

JADE es un entorno que simplifica la implementación de sistemas multiagente mediante una capa de soporte (middle-ware) que respeta las especificaciones FIPA y con un conjunto de herramientas para el desarrollo y debugging. La plataforma puede ser distribuida en varias máquinas (las cuales no necesitan compartir el mismo sistema operativo) y la configuración puede ser controlada mediante una interface gráfica remota. La configuración puede incluso ser cambiada en tiempo de ejecución moviendo agentes de una máquina a otra, cuando es necesario.

De todas las mencionadas con anterioridad se selecciona JADE como marco de trabajo más idóneo; pues su arquitectura de comunicación ofrece mensajes flexibles y eficientes, mientras que JADE crea y maneja una cola de mensajes ACL entrantes; los agentes pueden acceder a su cola mediante una combinación de varios modos: blocking, polling, timeout y patternmatching. El modelo de comunicación de FIPA ha sido implementado completo y sus componentes han sido claramente distinguidas y completamente integradas: protocolos de interacción, ACL, lenguaje de contenido, esquemas de codificación, ontologías y finalmente protocolos de transporte. El mecanismo de transporte, en particular, es como un camaleón debido a que se adapta a cada situación, seleccionando transparentemente el mejor protocolo disponible entre Invocación de métodos Remotos (RMI) de Java, notificación de eventos e Protocolo de Internet Inter-ORB (IIOP). Otros protocolos pueden ser fácilmente agregados. Integraciones de Protocolo de Transferencia de Correo Simple(SMTP), HTTP y Protocolo de aplicación inalámbricas (WAP) ya están planificadas para ser incorporadas. Muchos de los protocolos de interacción de FIPA ya están disponibles y pueden ser instanciados después de definir la dependencia de la aplicación de cada estado del protocolo.

Segunda Vida (SL) y ontologías de manejo de agentes están implementados, así como el soporte para lenguajes de contenido y ontologías definidos por el usuario que pueden ser implementadas, registradas con los agentes y automáticamente utilizadas por el framework.

1.4 Lenguaje de programación.

➤ JAVA:

Java es un lenguaje de desarrollo de propósito general, con él se puede realizar todo tipo de aplicaciones. Lo que lo diferencia de otros lenguajes es que sus aplicaciones pueden ejecutarse en cualquier sistema operativo, solo requiere de la instalación correcta de sus máquinas virtuales entre las ventajas de este lenguaje se destacan:(Tomado parcialmente de (Sánchez 2003))

- Elimina los punteros lo que aumenta su seguridad.
- Es totalmente orientado a objetos (POO).
- Implementa de forma nativa excepciones.
- Es interpretado, es un pequeño programa el que interpreta a JAVA y no el Sistema Operativo.
- Es multihilo, permite al ordenador realizar varias tareas.
- Es dinámico, los objetos se cargan en memoria cuando son necesarios.

➤ SQL:

El SQL es el lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales. Es un lenguaje declarativo: sólo hay que indicar qué se quiere hacer. En cambio, en los lenguajes procedimentales es necesario especificar cómo hay que hacer cualquier acción sobre la base de datos. El SQL es un lenguaje muy parecido al lenguaje natural; concretamente, se parece al inglés, y es muy expresivo.(Escofet 2007)

1.5- Entorno de desarrollo integrado (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).(Díaz 2014)

➤ Netbeans:

NetBeans es un proyecto de código abierto que 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 Aplicaciones de Programación de Interfaces (APIs) de NetBeans y un archivo especial (manifest file) que lo identifica como módulo.(Díaz 2013)

1.6- Gestor de Base de datos.

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación.

➤ SQLite:

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. Sqlite se puede usar en modo ventana de comandos (Shell) o embebido en aplicaciones de código (casi cualquier lenguaje de programación directamente o indirectamente) por ejemplo c, c++, bash etc. En aplicaciones se puede utilizar con OpenOffice mediante la aplicación de los drivers ODBC adecuados. Sqlite es un motor que trabaja embebido con la aplicación que lo use (en el servidor, en programas escritos por nosotros, etc) y otra de sus ventajas es la portabilidad.

Sqlite 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.(Aguiar 2006)

1.7- Emoción a revertir: La Ira.

Para obtener una panorámica completa del camino a recorrer para el desarrollo de esta aplicación de software es necesario reconocer y definir cuál emoción debe revertir, y cómo lograrlo.

1.7.1- Acercamiento a la Ira.

La ira como se definió y caracterizó con anterioridad es un sentimiento destructivo que puede dañar ampliamente todo aquello que se encuentre en su radio de acción; por esta razón la psicología se dedicó a definir los tratamientos más efectivos a aplicar en un individuo que está bajo los efectos de la ira.

1.7.2- Tratamiento psicológico para la ira.

- **Diálogo interno:** Introspección
- **Autoafirmación positiva:** ver las cosas de forma más positiva.
- **No darle vueltas al asunto.** Dejarlo estar. Rechazo / Evitación. Ante una carga emocional, ayuda pensar: "NO" y "PUNTO". Eso no significa "no debo pensar / hablar sobre esto, o no hacerlo". Significa sólo "no", eso es todo, y punto final.
- **La interrogación cognitiva:** Superación orientada a una nueva valoración. Cuestionar la objetividad de los pensamientos derrotistas y autodestructivos. Identificar situaciones que generan preocupación, antes de que se inicie la espiral de ansiedad
- **Reestructuración cognitiva:** Cambiar pensamientos negativos por otros positivos.
- **Terapia emocional (reactiva) y educación emocional (proactiva):** Educar el afecto. Prevenir para maximizar las tendencias constructivas y minimizar las destructivas.
- **Relajación:** a diario, para estar preparados cuando surja. No sirven si sólo se utiliza en el momento en que se genera.
- **Actividad física:** deportes, pasear,...etc.
- **Diversiones:** distraerse con otras actividades "alegres", amigos, lectura, música, espectáculos, ir de vacaciones,...etc.
- **Placeres sensuales:** manjares, baños, música, sexo, comprar (se) obsequios, ir de compras,...etc.

- **Ayudar a los demás:** implicarse en actividades sociales, voluntariado,...etc.

Hasta el momento se ha mirado el tratamiento para esta emoción desde el punto de vista psicológico; pero a continuación se realizará un acercamiento al tratamiento desde la perspectiva de la programación afectiva.

1.7.3- Tratamiento desde la computación afectiva para la ira.

Teniendo en cuenta todo lo abordado hasta este punto se definieron los tratamientos a aplicar por el SMA para revertir la ira en el individuo como a continuación se describen:

- **Activar el reproductor:** El reproductor deberá activarse con el sonido de una música suave de preferencia del usuario.
- **Presentar mensajes:** Se enseñará al usuario mensajes cortos que animen a la reflexión del usuario.
- **Presentar Imágenes:** Se presentarán imágenes fuertes que encarnen las grandes barbaries que se pueden llegar a realizar bajo los dominios de esta emoción negativa.
- **Reproducir videos:** Se presentarán videos que traten temas sobre relajación, expliquen métodos para calmar la ira, y muestren el consejo de algunos especialistas.

Conclusiones:

En este capítulo se definió el término agente y sus características, se produjo un acercamiento a los SMA, se mencionaron las principales metodologías de diseño aplicables a ellos, así como las plataformas más conocidas para su implementación, también se produjo una aproximación a la ira, explicando cual es el tratamiento aplicado por la psicología para tratar la emoción, y cuál sería el que una computadora podría aplicar en el mismo caso.

El análisis de lo anterior permitió seleccionar como metodología para el diseño PASSI, pues a pesar de tener algunas desventajas, es la única de las analizadas que recorre todo el ciclo de vida de un SMA, además es compatible

con el IDE Netbeans el cual es capaz de soportar la plataforma JADE pues existe una total correspondencia entre ellos.

Se escogió Java como lenguaje de propósito general pues JADE es un marco de trabajo completamente implementado en Java que permite una integración segura y óptima con este lenguaje de programación, el cual es intrínsecamente orientado a objetos. Tiene una gran funcionalidad gracias a sus librerías (clases),

Como Gestor de base de datos se seleccionó SQLite el cual es una librería en proceso que implementa una base de datos sql propia sin necesidad de servidor o configuración, por esta razón se seleccionó este lenguaje de programación para construir la base de datos.

Capítulo 2: Diseño del SMA

Introducción:

Este capítulo está orientado a la construcción y diseño del SMA utilizando la metodología PASSI, se recorrerán las 2 primeras fases de vida del SMA con sus niveles correspondientes. La herramienta PASSI Tool Kit, plugin del Rational Rose será la encargada de la construcción de sus artefactos.

2.2- Nivel 1: Modelo de requerimiento del sistema:

Este modelo se confecciona en términos de agencia y propósitos. El desarrollo del mismo involucra las siguientes fases:

2.2.1- Descripción del entorno.

La fase de descripción del entorno lleva a la visión del sistema a su nivel máximo de abstracción. Es una descripción funcional utilizando un diagrama de caso de uso. Teniendo en cuenta las características de este SMA descritas con anterioridad, se considera el siguiente diagrama (Figura 3) como un único elemento integrado, el cual está compuesto por dos actores. Es primero será el estudiante, que representa la interacción del sistema con el entorno. Es segundo de ellos lo constituye la PC, este representa esquemáticamente las diferentes fuentes de información (Base de datos, repositorio), necesarias para el funcionamiento correcto del sistema.

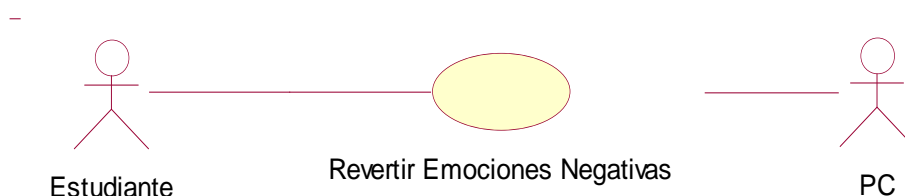


Figura 3. Diagrama de descripción del entorno.

2.2.2- Descripción del Dominio:

En esta fase se realiza una descripción funcional del sistema utilizando Diagrama de Casos de Uso y estereotipos UML, es importante destacar que de los casos de usos especificados tendrá lugar la siguiente fase perteneciente a este nivel, la fase de Identificación de Agentes.

En esta fase presenciaremos 11 casos de uso:

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

Por la importancia que representan para el funcionamiento correcto del sistema se decide describir los Casos de Uso, a través del formato que se presenta,

Cargar Interfaz, Mostrar Información, Recibir Información, Validar Información, Autenticar, Crear Usuario, Actualizar Perfil, Obtener Emoción y tratar Ira.

Tabla 1 Descripción del CU. Cargar Interfaz.

Caso de Uso: Cargar Interfaz.		
Descripción:	Este caso de caso de uso tiene como objetivo levantar la Interfaz principal, mostrando las opciones a escoger por el estudiante.	
Comunicaciones:	Iniciador	Participantes
	1. Cargar Interfaz.	1. Mostrar Información

Tabla 2 Descripción del CU. Mostrar Información.

Caso de Uso: Mostrar Información.		
Descripción:	Este caso de caso de uso tiene como objetivo mostrar la información proveniente del sistema al estudiante, permitiendo su interacción con este.	
Comunicaciones:	Iniciador	Participantes
	1- Cargar Interfaz.	1- Mostrar Información

Tabla 3 Descripción del CU. Recibir Información.

Caso de Uso: Recibir Información		
Descripción:	Este caso de caso de uso tiene como objetivo recibir la información proveniente del estudiante e introducirla al sistema	
Comunicaciones:	Iniciador	Participantes
	1- Recibir Información	1- Validar Información

Tabla 4: Descripción del CU. Validar Información.

Caso de Uso: Validar Información		
Descripción:	Este caso de caso de uso tiene como objetivo recibir la información proveniente del estudiante e introducirla al sistema los datos estén correctos.	
Comunicaciones:	Iniciador	Participantes
	1-Recibir Información	1-Validar Información

Tabla 5: Descripción del CU. Autenticar.

Caso de Uso: Autenticar		
Descripción:	Este caso de caso de uso tiene como objetivo que el estudiante se autentique al iniciar su interacción 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-Recibir Información 2-Autenticar	1-Autenticar 2-Mostrar Información.

Tabla 6: Descripción del CU. Actualizar Perfil.

Caso de Uso: Actualizar Perfil		
Descripción:	Este caso de caso de uso tiene como objetivo que el estudiante tenga la posibilidad de cambiar sus datos, y sus preferencias.	
Comunicaciones:	Iniciador	Participantes
	1-Recibir Información 2-Actualizar Perfil.	1-Actualizar Perfil 2-Mostrar Información.

Tabla 7: Descripción del CU. Crear Usuario.

Caso de Uso: Crear Usuario		
Descripción:	Este caso de caso de uso tiene como objetivo que el estudiante tenga la posibilidad de introducir sus datos en el sistema, especificar sus preferencias, creando su propio perfil.	
Comunicaciones:	Iniciador	Participantes
	1-Recibir Información 2-Crear Usuario	1-Crear Usuario 2-Mostrar Información.

Tabla 8 Descripción del CU. Obtener Emoción.

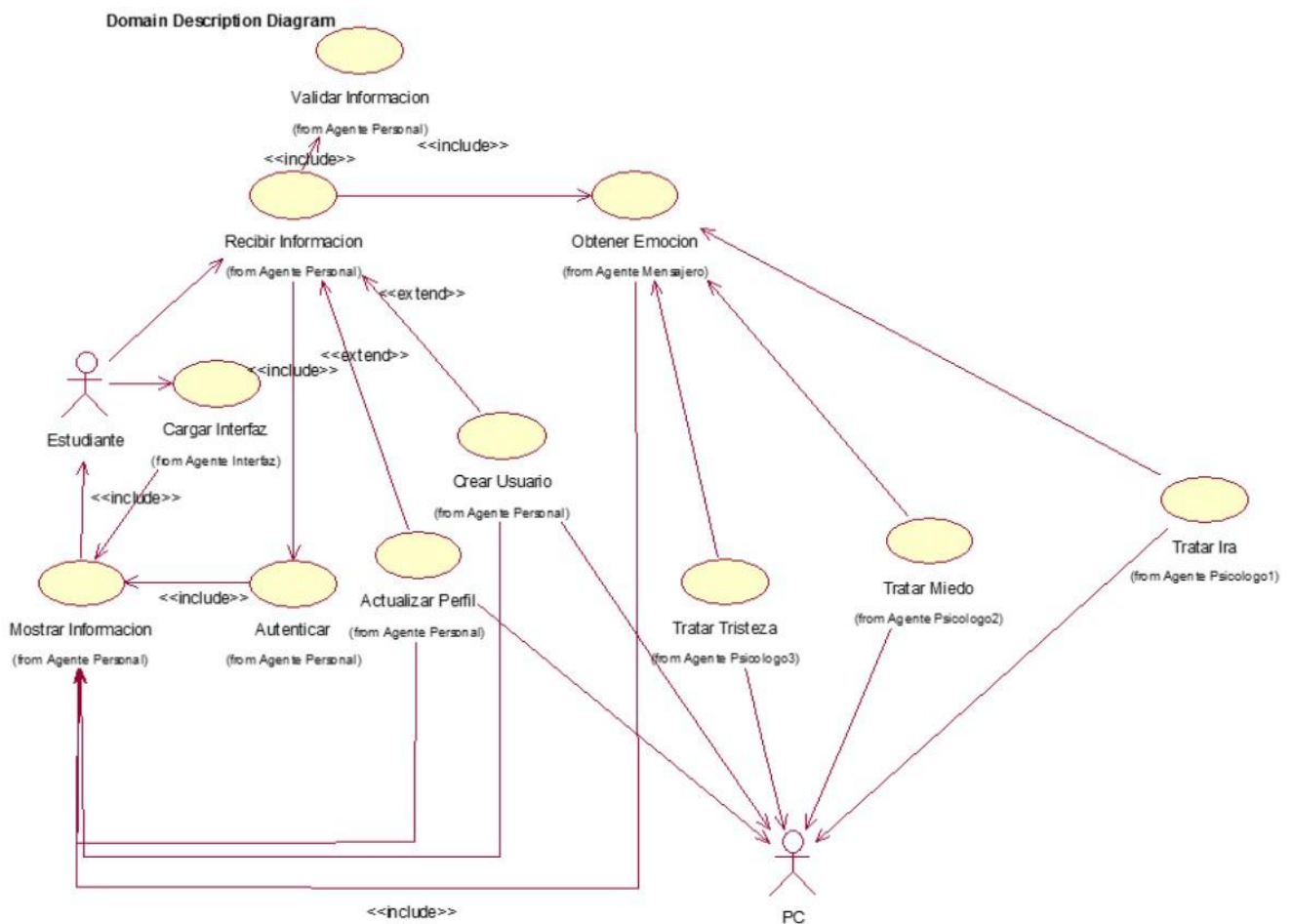
Caso de Uso: Obtener Emoción		
Descripción:	Este caso de caso de uso tiene como objetivo que el estudiante tenga la posibilidad de seleccionar la emoción que presenta en ese momento	
Comunicaciones:	Iniciador	Participantes
	1-Recibir Información 2-Obtener Emoción	1-Obtener Emoción 2-Tratar Tristeza

En la siguiente imagen se muestra el diagrama de descripción del dominio. (Figura 4)

2.2.3- Identificación de agentes.

PASSI haciendo uso del PassiTool Kit a partir del nivel anterior, genera el Diagrama de Identificación de Agentes. En él los agentes son vistos como paquetes UML, donde, en cada uno de ellos, se agrupan los Casos de Uso que pertenecen a cada agente, siendo estos las funcionalidades específicas de cada agente. (Figura 5)

En esta fase se identificaron 6 agentes; por la importancia que estos presentan para el correcto funcionamiento del sistema, los cuales serán mencionados a



continuación con los diferentes casos de uso que engloban.

Figura 4: Diagrama de descripción del dominio.

- **Agente Interfaz:**

El Caso de Uso que incorpora este agente es.

CU. Cargar Interfaz:

Este agente es el encargado de presentar la pantalla principal del sistema, dando la posibilidad de que el estudiante interactúe con el sistema.

- **Agente Personal:**

CU. Recibir Información

CU. Mostrar Información

CU. Validar Información

CU. Crear Usuario

CU. Actualizar Usuario.

CU. Autenticar.

Este agente es el encargado de la gestión del usuario. Es propuesto como moderador entre el sistema y los recursos externos donde se encuentra almacenada la información.

- **Agente Mensajero:**

Incorpora el Caso de Uso:

CU. Obtener emoción.

Este agente es el encargado de recibir desde personal la emoción que presenta el usuario, de acuerdo a ella, él deberá despertar al agente psicólogo que tratará dicha emoción.

- **Psicólogo 1:**

Incorpora el Caso de Uso:

CU. Tratar Ira.

Este agente será el encargado de buscar en la PC un tratamiento adecuado para la emoción y enviarlo hasta el usuario.

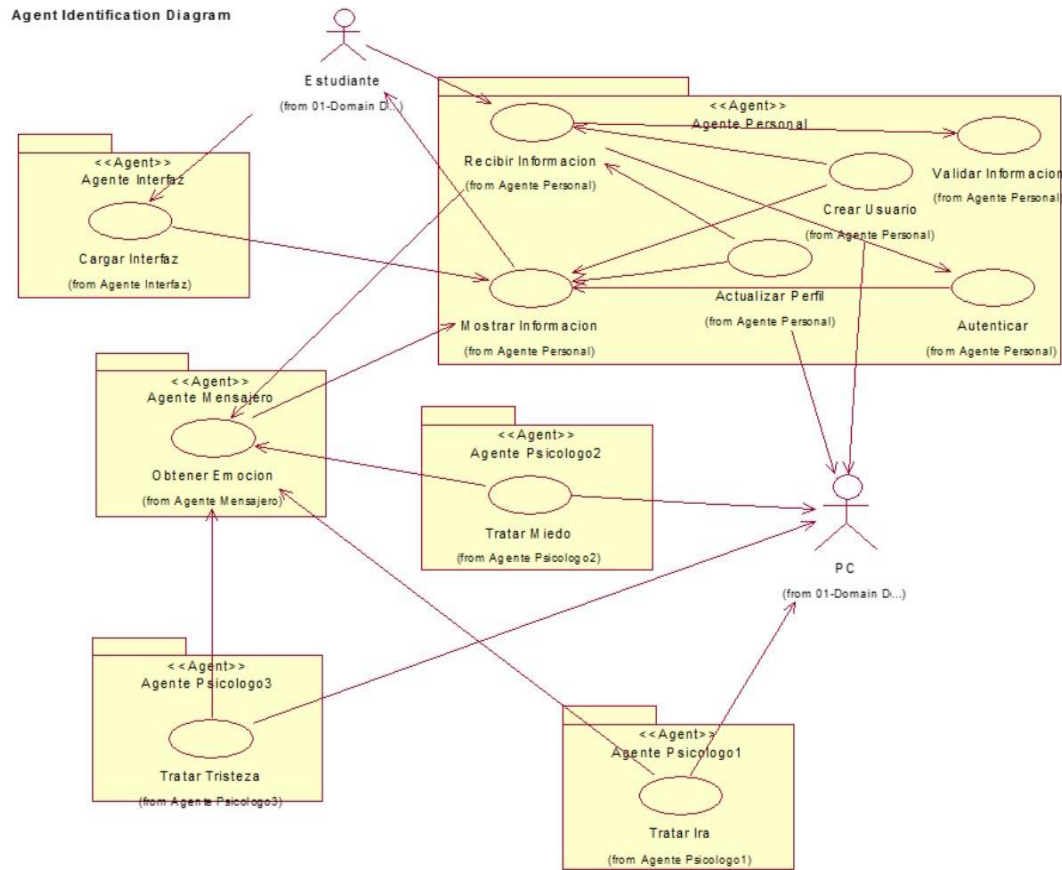


Figura 5: Diagrama de identificación de agentes.

2.2.4- Identificación de Roles:

En esta fase se definen las responsabilidades de cada agente a través de escenarios específicos para cada rol a través de Diagramas de Secuencias, que no son más que caminos que definen las diferentes conductas del sistema. (Rojas 2015) A partir de esta fase, solo se especificarán los diagramas relacionados con el Agente Psicólogo 1 y el Agente Personal, el resto se podrá encontrar en los anexos.

Los escenarios identificados fueron:

- Gestionar Usuario (Figura 6)

- Gestionar Emoción. (Figura 7)

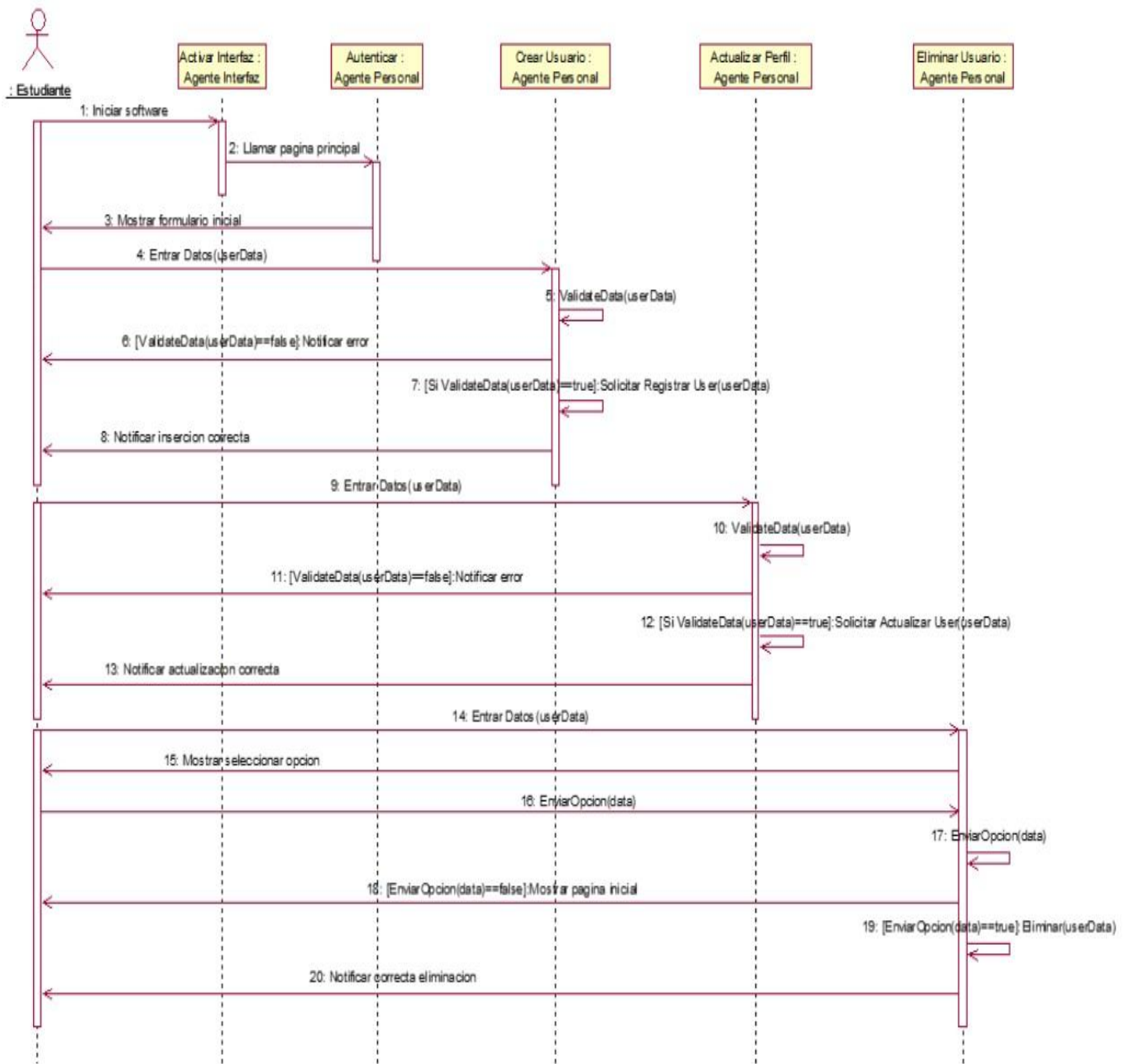


Figura 6: Diagrama de secuencia del escenario: Gestionar Usuario.

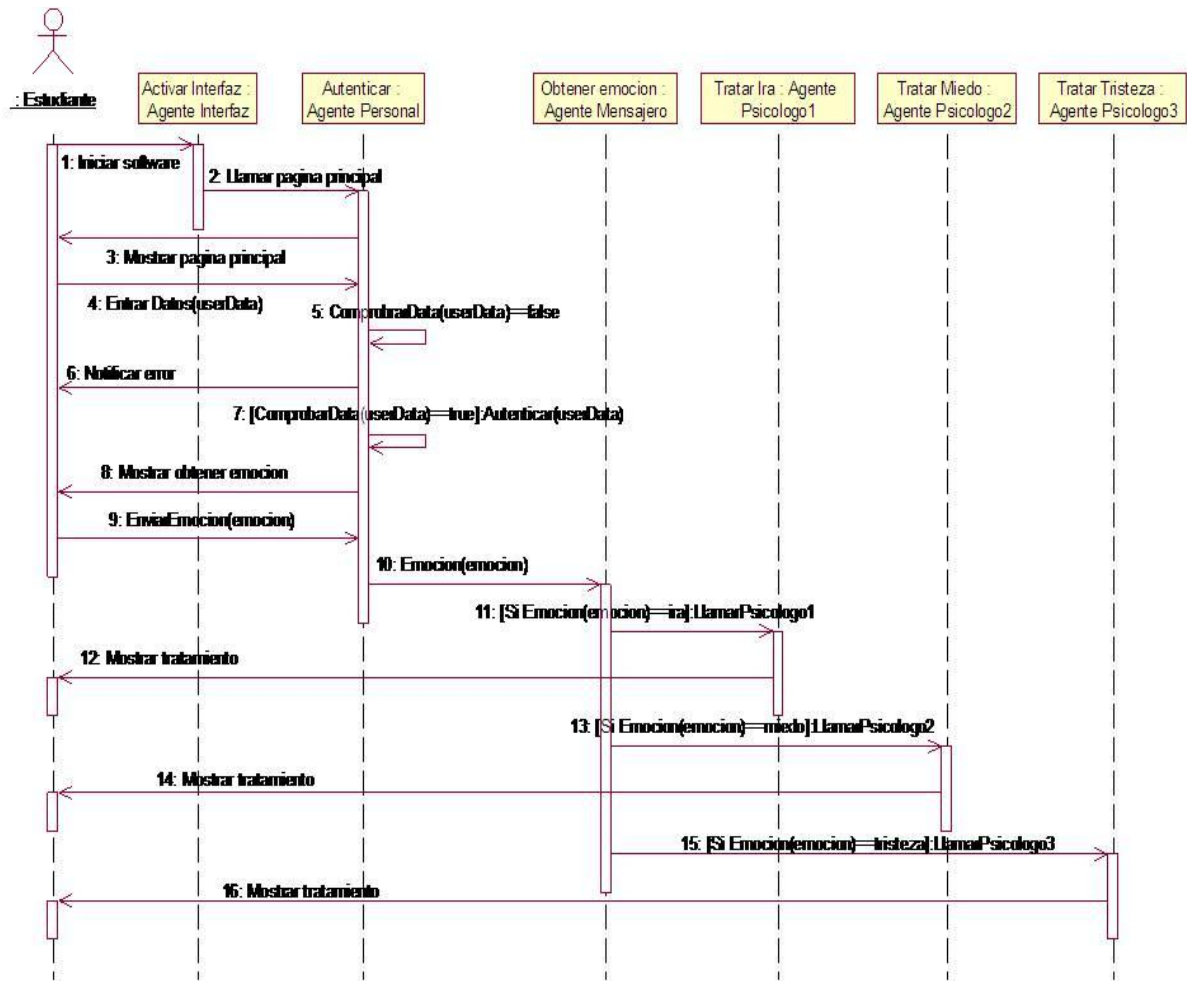


Figura 7: Diagrama de secuencia del escenario: Gestionar Emoción.

Identificación de Roles más relevantes de los agentes Propuestos:

Agente Interfaz:

- Rol Activar Interfaz

Agente Personal:

- Rol Autenticar
- Rol Crear Usuario

- Rol Actualizar Perfil
- Rol Eliminar Usuario

Agente Mensajero:

- Rol Obtener Emoción

Agente Psicólogo 1:

- Rol Tratar Ira.

Tabla 9: Descripción del escenario: 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 Actualizar Perfil CU Recibir Información CU Autenticar
Roles Asociados	Descripción
Rol Activar Interfaz (Agente Interfaz)	Este rol es el responsable de la comunicación entre los usuarios y el sistema. Sus principales actividades son receptionar, mostrar y validar información.
Rol Crear Usuario (Agente Personal)	Este rol es el responsable de la creación de un nuevo usuario del sistema. Su principal actividades guardar los datos proporcionados por el usuario en la Base Datos del Sistema.
Rol Actualizar Perfil (Agente Personal)	Este rol es el responsable de actualizar los datos de los usuarios y guardar los cambios en la Base de Datos del

	Sistema.
Rol Eliminar Usuario (Agente Personal)	Este rol es el responsable de eliminar de la Base de datos del sistema un usuario determinado.

Tabla 10: Descripción del escenario: Gestionar Emoción.

Escenario: Gestionar Emoción.	
Descripción	Este escenario muestra el curso normal que sigue el sistema al obtener la emoción y aplicar un tratamiento para revertirla.
CU Asociados:	<ol style="list-style-type: none"> 1. CU Autenticar 2. CU Obtener Emoción. 3. CU Tratar Ira. 4. Tratar Tristeza. 5. Tratar Miedo.
Roles Asociados	Descripción
Rol Activar Interfaz (Agente Interfaz)	Este rol es el responsable de la comunicación entre los usuarios y el sistema. Sus principales actividades son recepcionar, mostrar y validar información.
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, y activar Agente Personal.
Rol Captar Emoción (Agente Mensajero)	Este rol es el encargado de recibir de personal 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

2.2.5- Especificación de Tareas.

En esta etapa se especifica a través de Diagramas de Actividades las capacidades de cada agente. Se obtiene de la fase Identificación de Roles, explorando todos los escenarios en que participa el agente. Es válido aclarar que de cada Diagrama de Identificación de Roles se obtiene una colección de tareas relacionadas, que de cuya agrupación resulta el Diagrama de Especificación de Tareas. Para cada agente propuesto se identificaron las tareas correspondientes (Figura 8.) y (Figura 9). (Ver anexo 1-4)

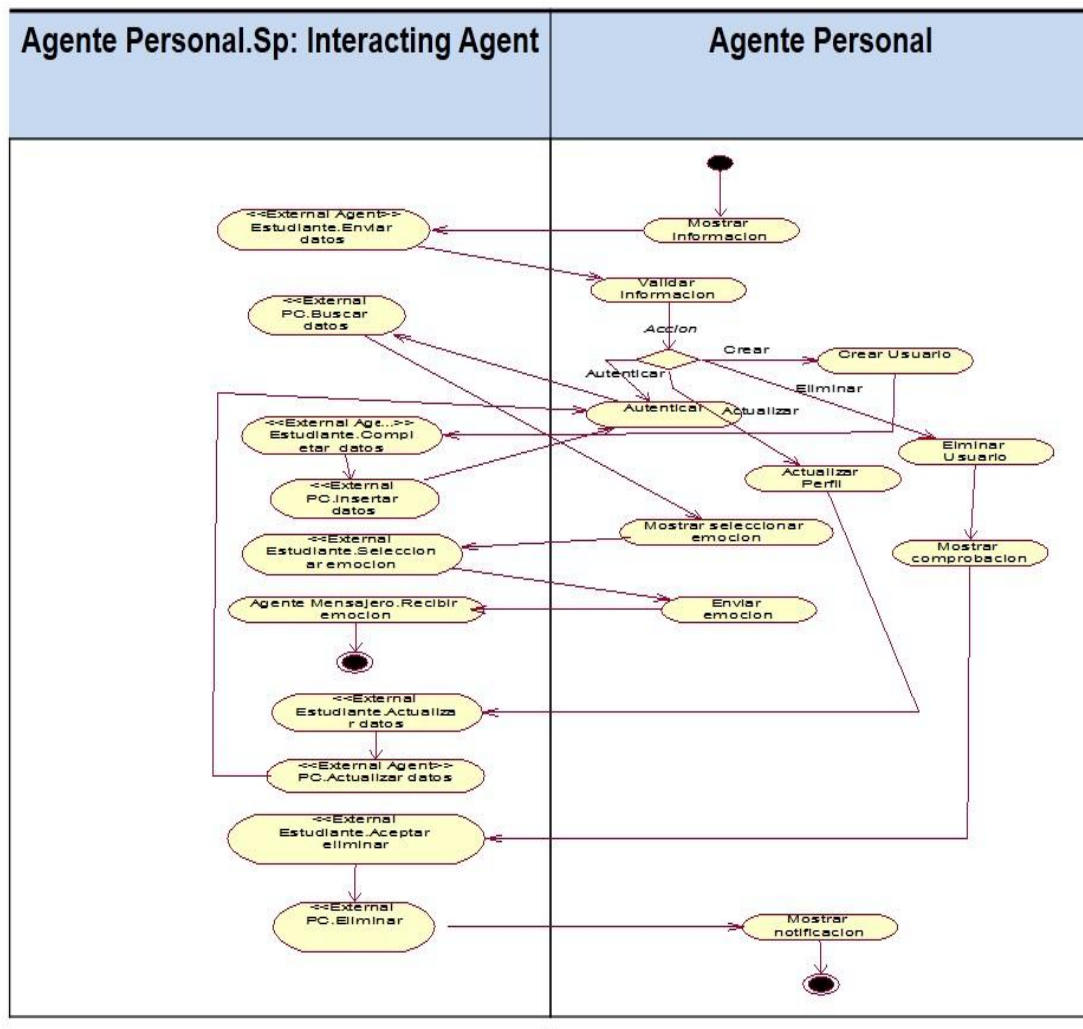


Figura 8: Diagrama de Actividad. Agente Personal.

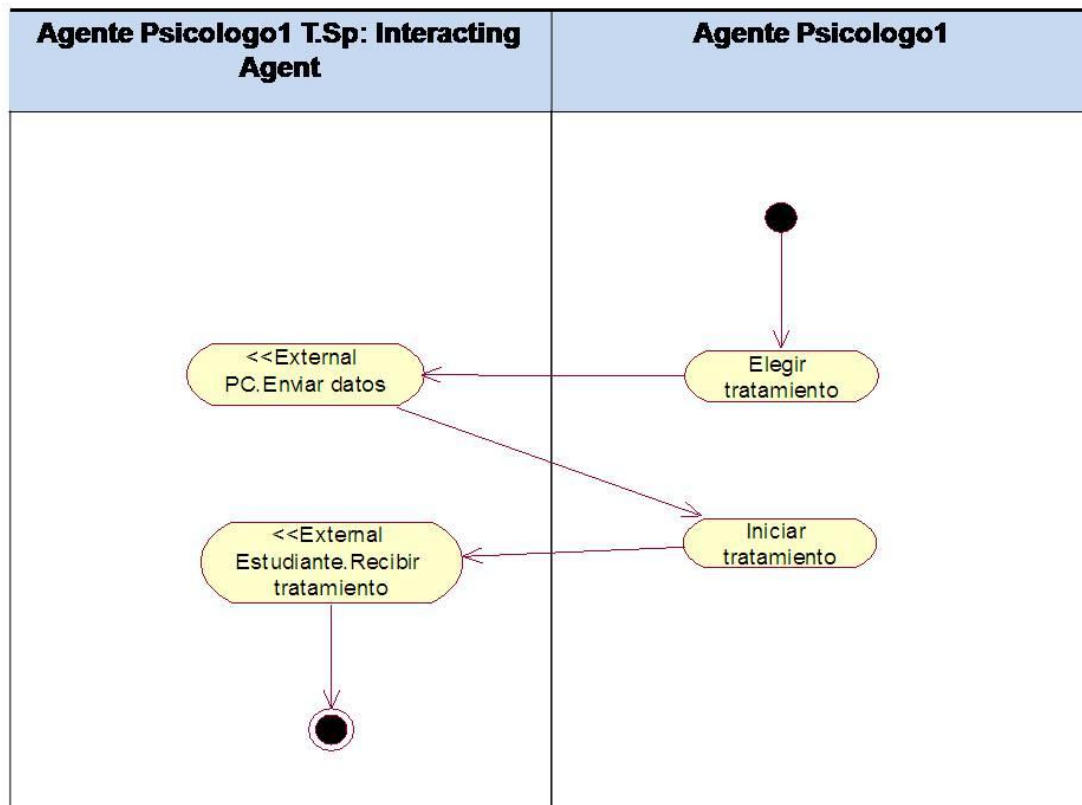


Figura 9: Diagrama de actividad. Agente Psicólogo1.

2.3- Nivel 2: Modelo de Sociedades de Agentes:

En este nivel se modelan las interacciones sociales y dependencias entre los agentes involucrados en la solución.

2.3.1- Descripción de la Ontología.

En esta fase se describe el conocimiento de cada agente y el tipo de comunicación que tendrá con otros agentes a través de los Diagramas de Clases.

2.3.1.1- Descripción de ontología del Dominio.

Se representan a las entidades a través de clases en el Diagrama de Ontología del Dominio, el cual no es más que una forma de describir el conocimiento,

muestra la manera de comunicación y la información que se generan en estas entidades.

En este diagrama (Figura 10) se definen una serie de clases:

- Conceptos: Son de color amarillo. Están representados con estereotipos UML. Describen el conocimiento que operan en los agentes.
- Predicados: Son de color azul. Definen cursos o posibles alternativas que se podrán ejecutar en el sistema.
- Acciones: Son de color blanco: Representan las funcionalidades de los conceptos.

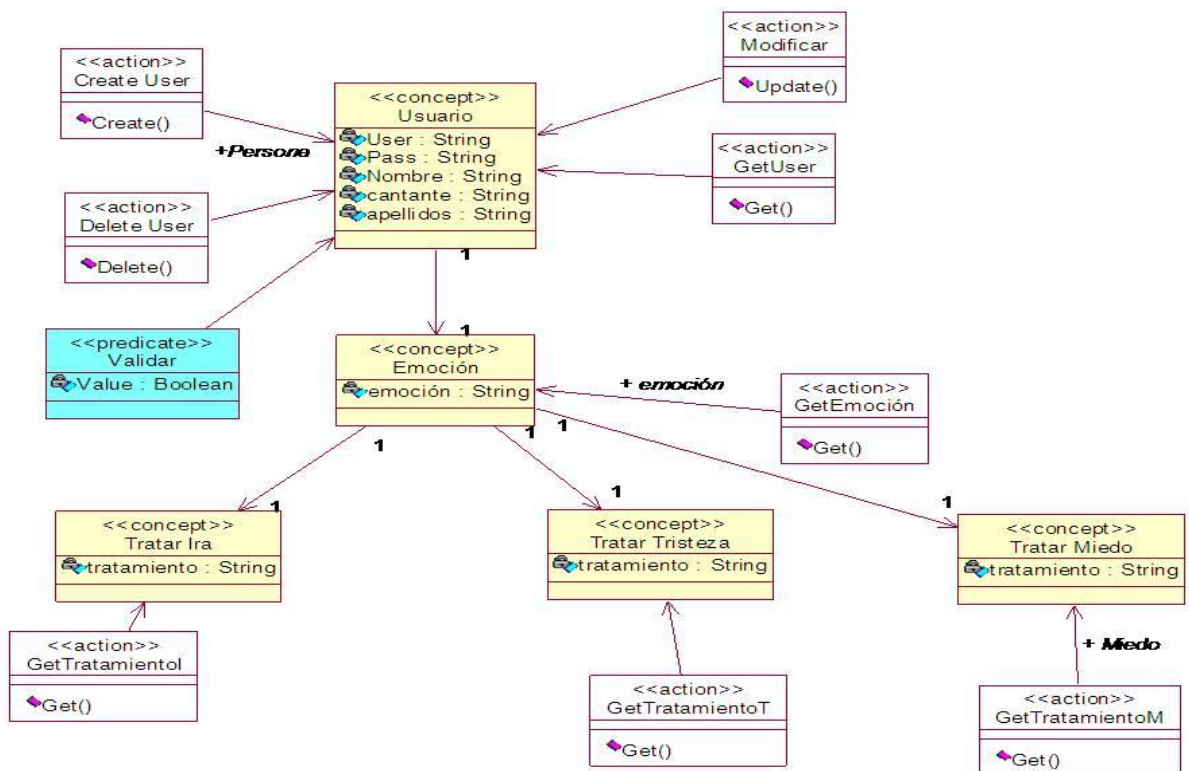
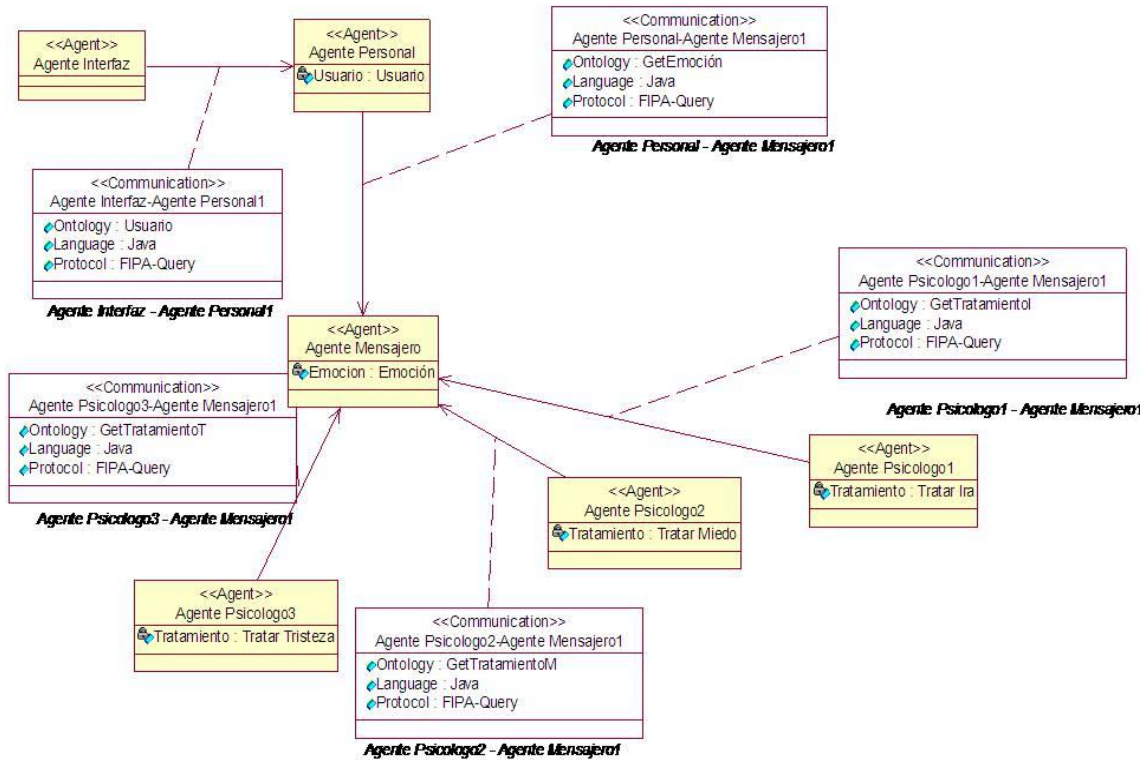


Figura 10: Diagrama de Descripción de Ontología del Dominio.

2.3.1.2- Descripción de Ontología de Comunicación.

Este diagrama (Figura 11) se enfoca en el conocimiento de los agentes y sus relaciones comunicativas. En necesario especificar en cada comunicación la



ontología, el lenguaje y el protocolo, estos últimos son estandarizados en su mayoría por FIPA(Rojas 2015).

Figura 11: Diagrama de Descripción de Ontología de Comunicación.

2.3.2- Descripción de Roles.

Esta fase es descrita a través de un Diagrama de Paquetes (Figura 12). En él existe una clase para cada rol, donde se le introducen las tareas específicas que él debe efectuar, así como se especificarán las dependencias y comunicación entre ellos.

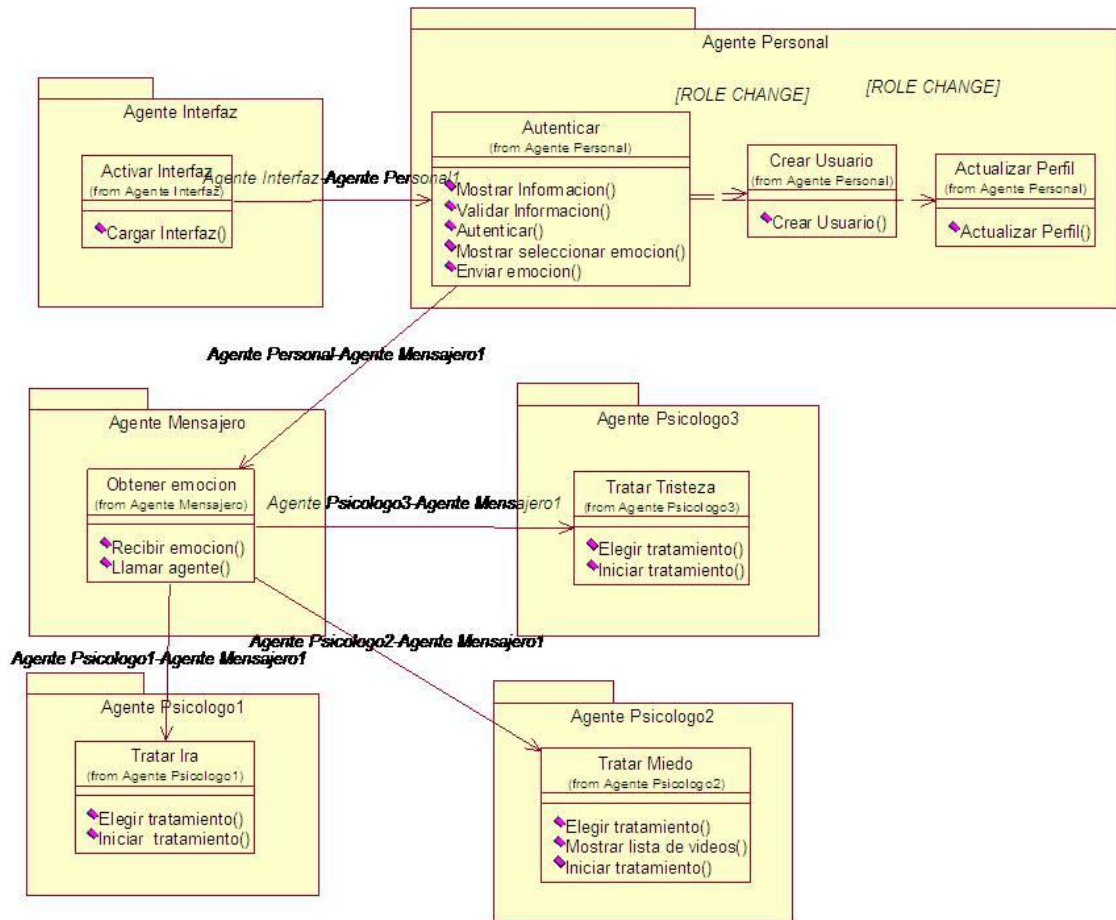


Figura 12: Diagrama de Descripción de Roles.

2.3.3- Descripción del Protocolo.

Esta fase especifica la gramática de cada protocolo de comunicación, que no es más que determinar la forma en que los agentes se comunicarán. Los protocolos seleccionados para modelar este proceso fueron el FIPA-QUERY y FIPA-REQUEST. El diagrama de secuencias correspondiente al primero de ellos es el siguiente (Figura 13).

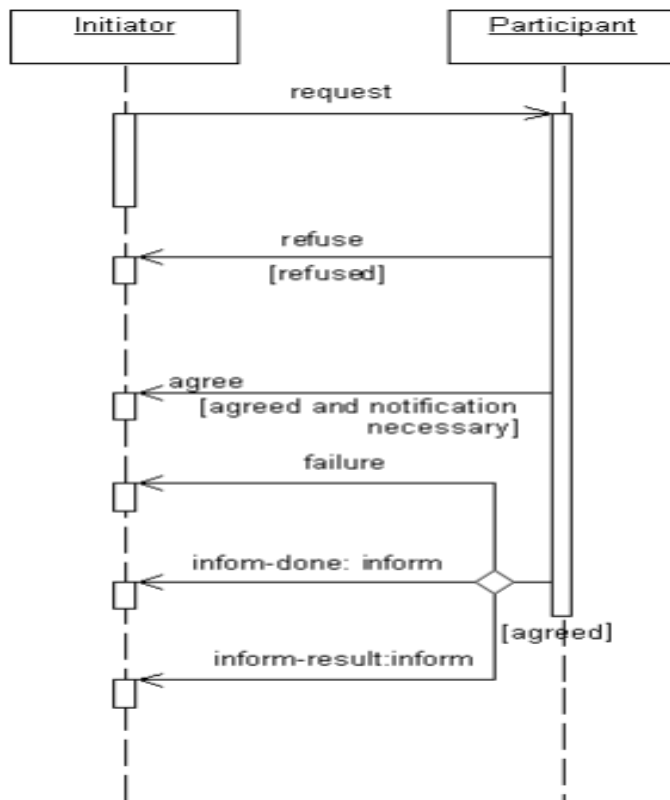


Figura 13: Diagrama de Descripción de Protocolo.

Conclusiones:

En este capítulo se desarrolló los 2 primeros niveles de la metodología PASSI. Al especificar el nivel 1 quedaron definidas las siguientes fases: descripción del entorno, descripción del dominio, identificación de agentes, identificación de roles, especificación de tareas.

En el nivel 2 las fases detalladas fueron las que siguen: Descripción de Ontología, Descripción de Roles, Descripción de Protocolos.

Capítulo 3: Implementación de la solución propuesta.

Introducción:

PASSI recorre todo el ciclo de vida del SMA. En este capítulo se transitará por los niveles 3, 4 y 5 de la mencionada metodología. Definiendo el modelo de Implementación del Agente, el Modelo de Código y el Modelo de Despliegue.

3.1- Nivel 3: Modelo de Implementación del Agente.

Este modelo proporciona una modelación de la arquitectura de la solución en términos de clases y métodos. Su desarrollo conlleva las siguientes fases:

3.1.1- Definición de estructura del Agente.

En esta fase se define la estructura de las clases del agente mediante el uso de Diagramas de Clases Convencionales el cual se divide en dos líneas: estructura del SMA y estructura del Agente simple.

3.1.2- Definición de Estructura del SMA:

Describe la estructura del SMA mediante un Diagrama de Clases donde cada clase representa un agente identificado en la fase de Identificación de Agentes. El conocimiento de cada agente puede representarse mediante compartimientos de atributos (como se vio en el Diagrama de Ontología), considerando el uso de compartimientos de operaciones para la identificación de las tareas del agente.(Pinto 2006)(Figura 14)

3.1.2- Definición de estructura del agente simple.

Durante esta fase se describe al agente como una estructura de clases, es decir la clase principal que es el agente y las clases internas que identifican sus tareas. Se utiliza un diagrama de clases para cada agente definiendo los atributos y métodos del mismo (Figura 15) y (Figura 16).

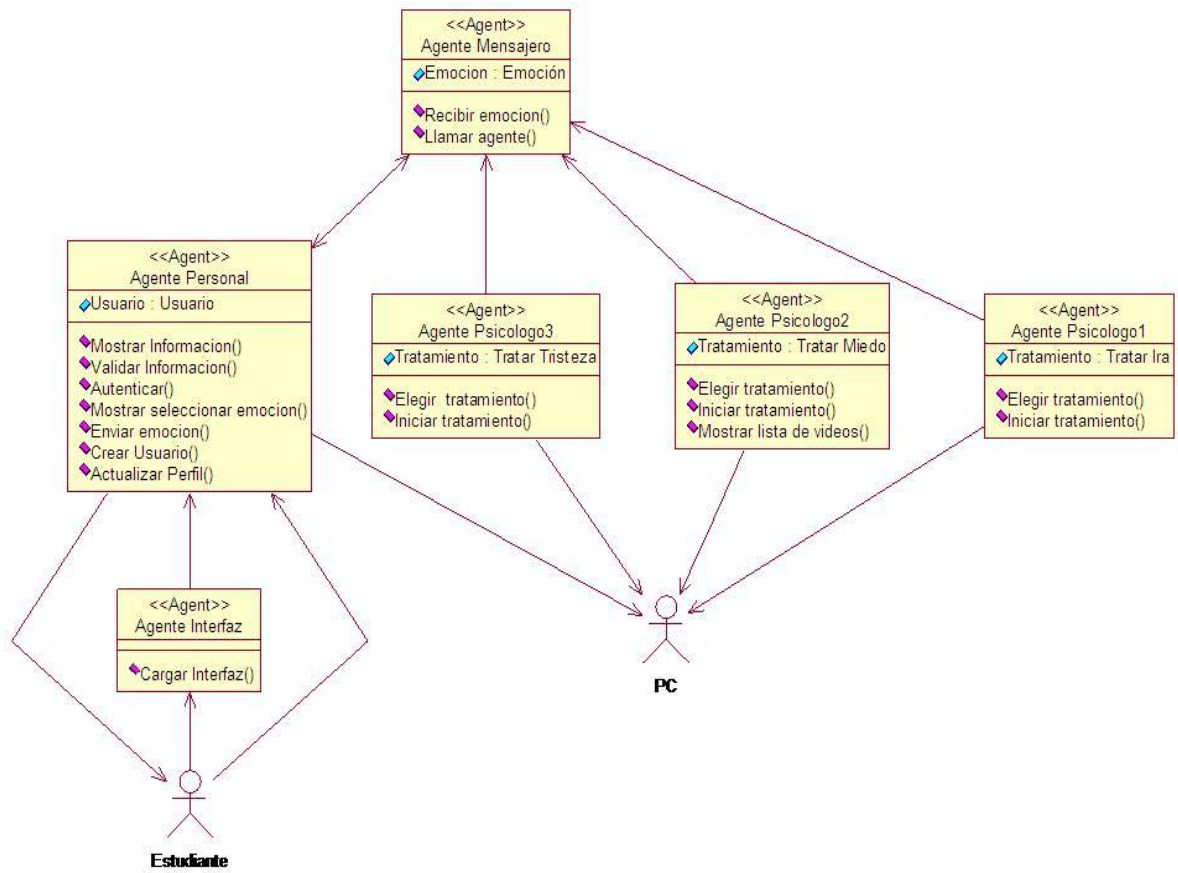


Figura 14: Diagrama de Descripción del SMA.

3.1.3- Descripción de Conducta del Agente.

Esta fase genera dos Diagramas de Clases, el que hace referencia al SMA refleja el flujo de eventos por invocación de los métodos e intercambio de mensajes, mientras que el del Agente, especifica los métodos anteriores.

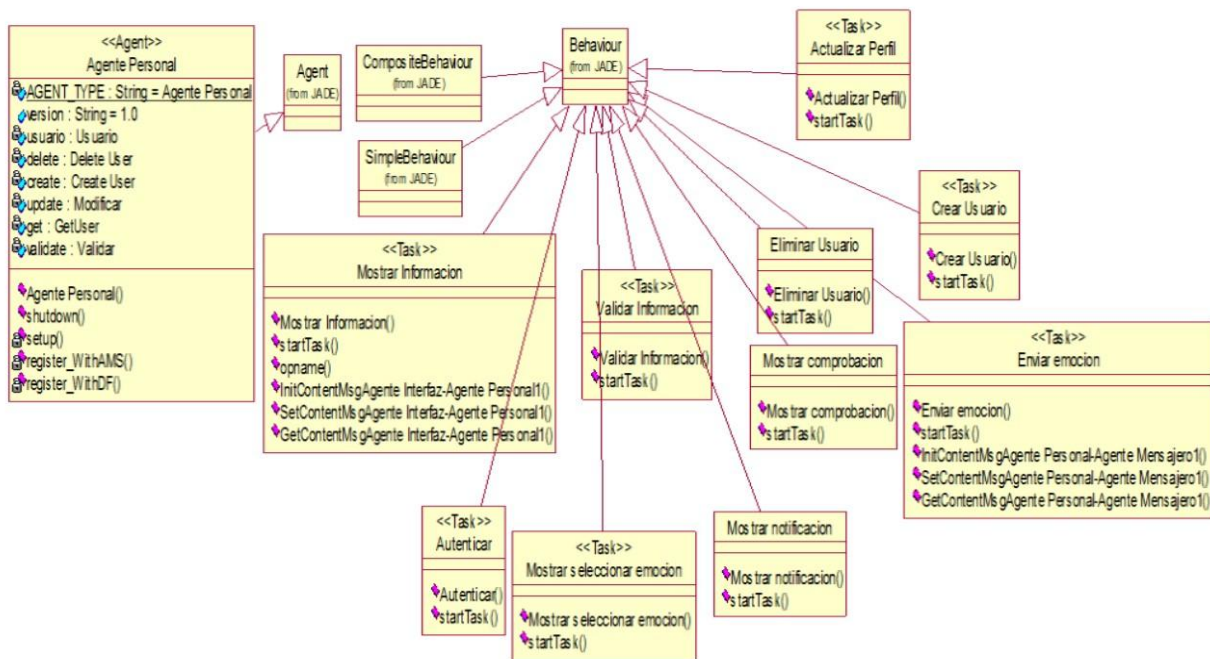


Figura 15: Diagrama de Descripción del agente simple: Personal

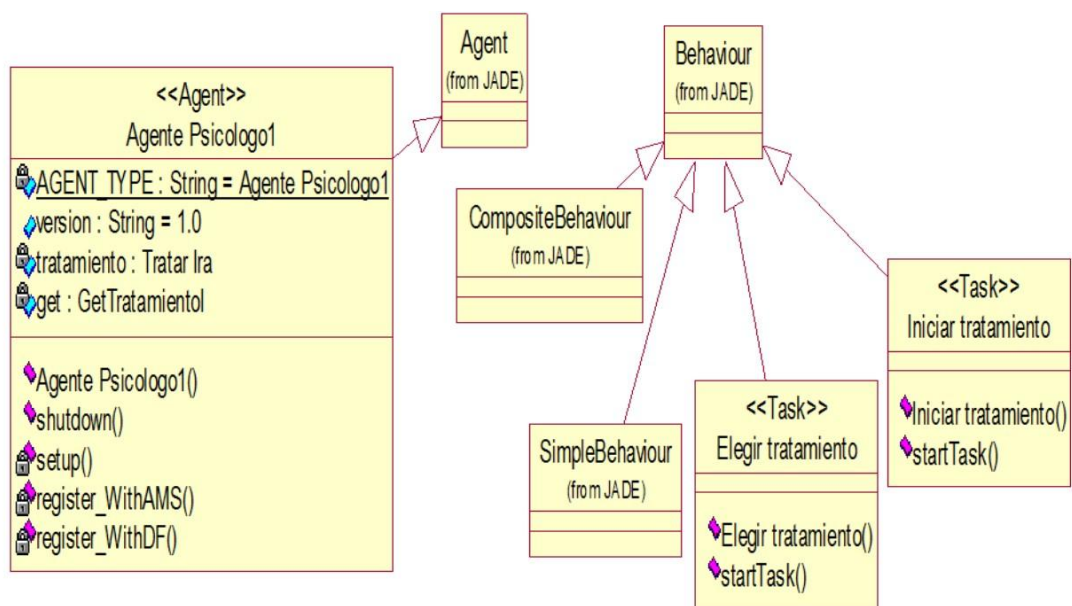
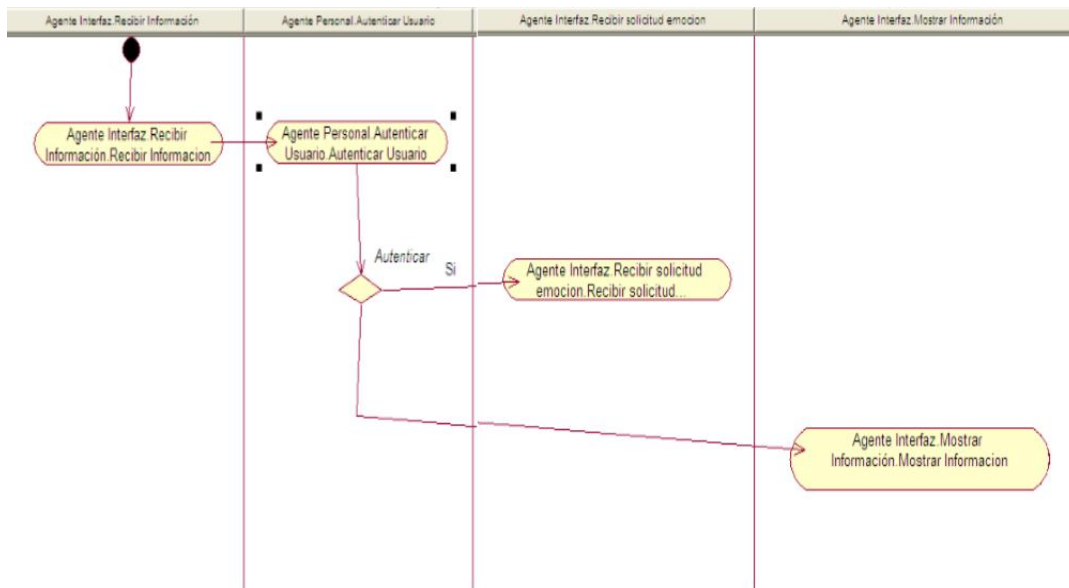


Figura 16: Diagrama de Descripción Agente Simple: Psicólogo1.

3.1.3.1- Descripción de Conducta del SMA.

Se muestra el flujo de eventos entre y dentro de los agentes y sus clases internas mediante uno o más diagramas de actividad. Se debe diseñar una



calle para cada agente y para cada tarea. (Figura 17 y 18)

Figura 17: Diagrama de Descripción de conducta del SMA. Escenario gestionar usuario.

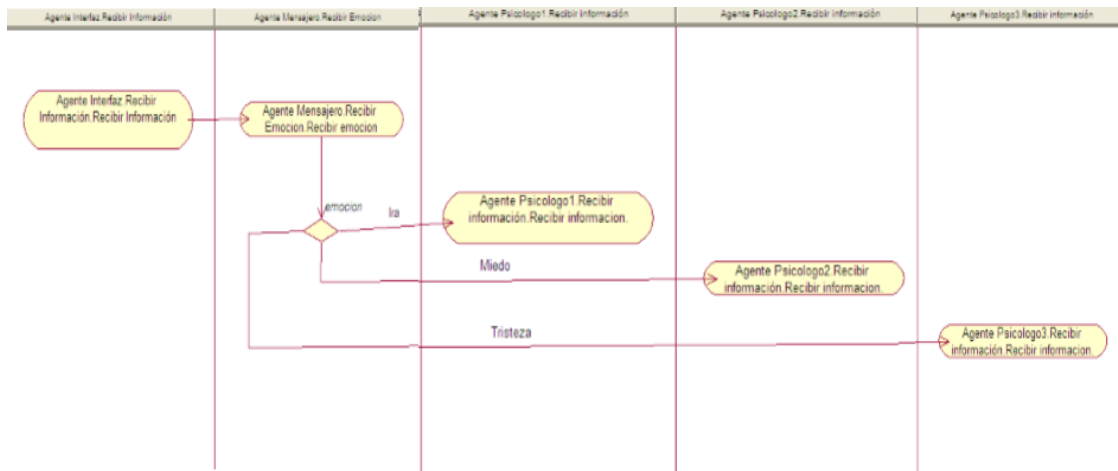


Figura 18: Diagrama de Descripción de conducta del SMA. Escenario gestionar emoción.

3.1.3.2- Descripción de conducta del agente simple.

Incorpora la implementación de los métodos introducidos en los Diagramas Definición Estructura del Agente, esta fase de la metodología no se corresponde con los objetivos propuestos para este Capítulo, pues la descripción detallada de los métodos de las clases de implementación identificados en las etapas anteriores supone un nivel de profundidad aceptable del nivel en cuestión.(Figura 19)

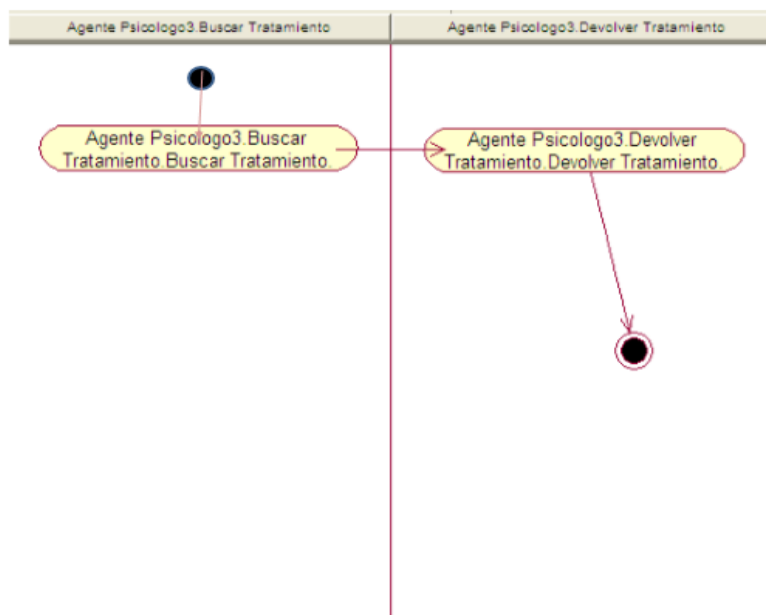


Figura 19: Diagrama de descripción del agente Psicólogo1.

3.2- Nivel 4: Modelo de Código.

3.2.1- Biblioteca Reutilizable de Código.

En esta fase se trata de reutilizar patrones existentes de agentes y tareas. Aquí se realiza la implementación de los agentes identificados, generándose para repetición uno una clase. Por su importancia para el proyecto se procede a mostrar los fragmentos de código generado por el PTK de los agentes personal y Psicólogo 1, los demás agentes podrán ser visualizados en los anexos.

➤ **Agente Personal:**

```
public class Agente Personal extends Agent {

private static String AGENT_TYPE = Agente Personal;
publicStringversion = 1.0;
private Usuario usuario ;
private Emoción emocion ;
private Modificar upDate ;
privateGetUsergetUser ;
privateCreateUsercreateUser ;
privateDeleteUserdelete ;
private Tratar Ira tratarIra ;
private Tratar Tristeza tratarTristeza ;
private Tratar Miedo tratarMiedo ;
private Validar validar ;

public void Agente Personal(String platform, String name, String
ownership) {
//insert here your code
}

public void shutdown() {
//insert here your code
}

private void setup() {
//insert here your code
}

private void register_WithAMS() {
//insert here your code
}

private void register_WithDF() {
```

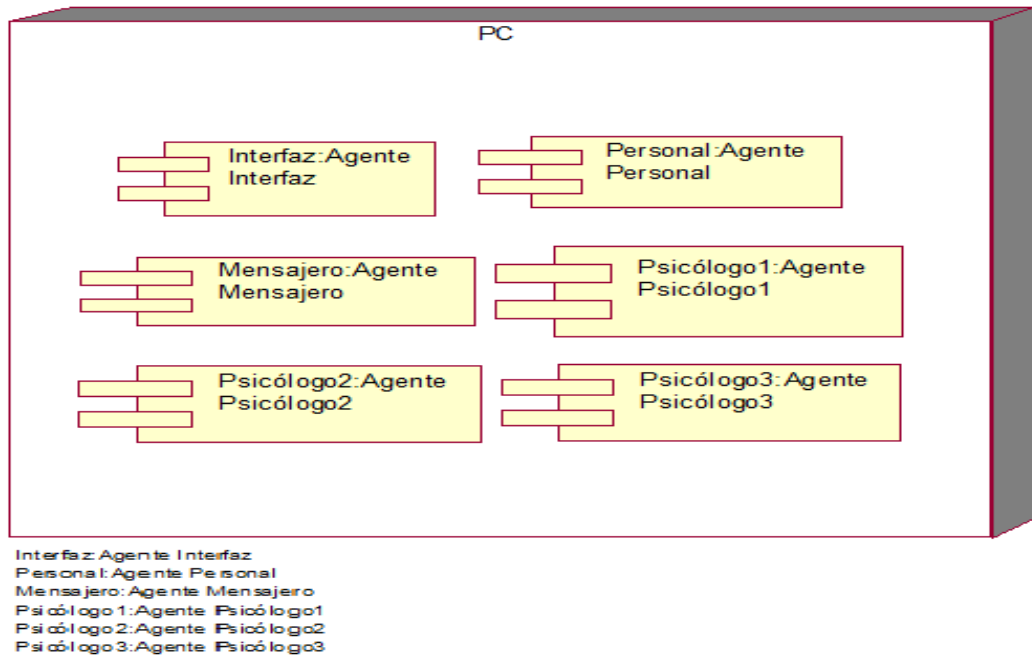
```
    //insert here your code  
}
```

➤ **Agente Psicólogo 1:**

```
public class Agente Psicologo1 extends Agent {  
  
private static String AGENT_TYPE = Agente Psicologo1;  
publicStringversion = 1.0;  
private Emoción emocion ;  
private Tratar Ira ira ;  
privateGetTratamientoIgetTratamientoI ;  
privateGetEmocióngetEmocion ;  
privateGetTratamientogetTratamiento ;  
  
public void Agente Psicologo1(String platform, String name,  
String ownership) {  
    //insert here your code  
}  
  
public void shutdown() {  
    //insert here your code  
}  
  
private void setup() {  
    //insert here your code  
}  
  
private void register_WithAMS() {  
    //insert here your code  
}  
  
private void register_WithDF() {  
    //inserthereyourcode  
}
```

3.2.2- Perfeccionamiento del código básico.

En esta fase se realiza la modificación de código fuente, ajustándose a los intereses específicos y funcionales de cada agente.



3.4- Nivel 5: Modelo de Despliegue.

Figura 20: Diagrama de Despliegue.

Este nivel constituye un modelo de la distribución del sistema, a través de unidades de procesamiento hardware y su integración entre las mismas incorpora una única fase (Figura 20).

3.5- Interfaces del sistema:

Como resultado final de la aplicación de la Metodología PASSI, la implementación del sistema a través de la plataforma Jade soportada sobre el IDE Netbeans, se pueden apreciar la ventana principal del SMA. (Figura 18)

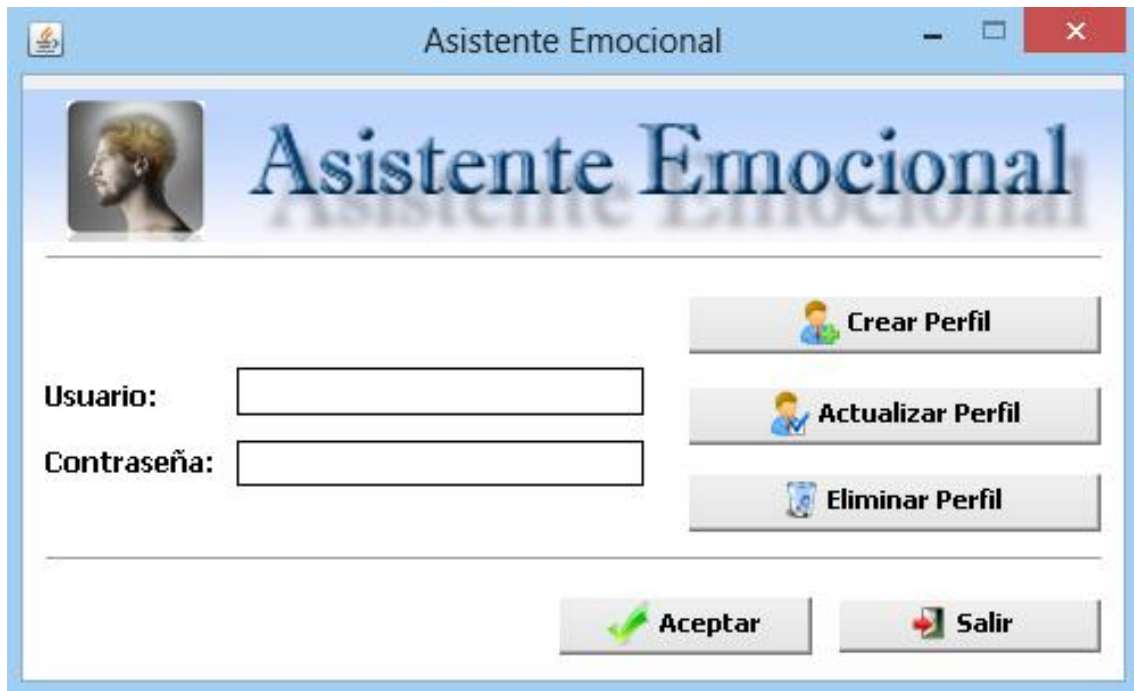


Figura 21: Ventana principal del SMA.

El usuario tendrá la posibilidad de escoger las siguientes opciones:

Crear Usuario: Una vez el usuario desee registrarse en el sistema y marque esta opción el agente Personal procederá levantar la siguiente ventana (Figura 23).

Eliminar Usuario: El estudiante tendrá la oportunidad de eliminarse él mismo del sistema, pero primeramente deberá ingresar su usuario y su contraseña. (Figura 24)

Actualizar Perfil: El estudiante podrá cambiar sus datos y modificar sus preferencias en esta ventana. (Figura 25)



Figura 22: Ventana del SMA. Crear Usuario.

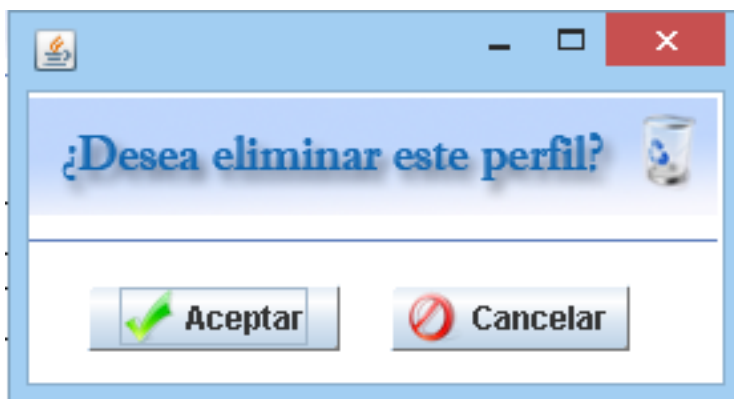


Figura 23: Ventana del SMA. Eliminar Usuario.

Una vez que el usuario haya introducido sus datos el agente Personal procederá a levantar una segunda interfaz con el propósito de captar la emoción que el usuario siente en ese momento. (Figura 26)

Asistente Emocional

Nombre:

Apellidos:

Usuario:

Contraseña:

Cantante:

Figura 24: Ventana del SMA. Actualizar Usuario.

¿Cómo te sientes hoy?


Estoy enojado Estoy triste Estoy temeroso

Me siento bien

Figura 25: Ventana del SMA. Obtener Emoción.


Una vez obtenido la emoción el sistema procederá a despertar al Agente Psicólogo1, el cual será el encargado de ejecutar uno de los 4 tratamientos implementados para revertir la Ira en el estudiante. (Figura 27)

Reproductor



Asistente Emocional

Controla tu carácter,
porque el enojo
es el
distintivo
de los necios.
Eclesiastés 6:9



◀ Atrás ▶ Adelante ▶

IMG



LA IRA HACE QUE DAÑES A LOS QUE TE RODEAN



Asistente Emocional

▶ Play

■ Stop

Conclusiones

En este capítulo se desarrolló los niveles 3, 4 y 5 de la metodología PASSI, de las fases del nivel 3 desarrolladas se puede hacer mención a: Definición de Estructura de SMA y del Agente, Descripción de Conducta del SMA y del agente.

Entre las fases del nivel 4 llevadas a cabo se encuentran: Reutilización del código y perfeccionamiento del código básico. Mientras que en el nivel 5 se desarrolló la fase de Configuración de despliegue.

Estos niveles permitieron definir la estructura del SMA y de cada agente simple definiendo sus atributos y métodos, de la misma manera se obtuvo una biblioteca reutilizable de código que funciona como base para la implementación del sistema, así mismo, se construyeron todas las ventanas necesarias para establecer la comunicación del usuario con el sistema.

Conclusiones:

1. El estudio de los fundamentos teóricos metodológicos, demostró que un SMA era perfectamente capaz de resolver el problema al integrar diferentes mecanismos con el objetivo de poder aplicar correctamente los diferentes tratamientos establecidos por los psicólogos.
2. Las diferentes fases y niveles que plantea PASSI permitieron realizar la documentación del SMA y en especial, el modelo de código permitió la generación de código automático lo facilitó el proceso de implementación.
3. A pesar de que existen varias plataformas para desarrollar el SMA, Jade resultó ser el middle-ware adecuado para el desarrollo de esta investigación el cual al integrarse con netbeans como IDE permitió crear el Asistente Emocional, de manera tal que se pueda integrar a cualquier proyecto.

Recomendaciones:

- ✓ Se recomienda establecer un orden de prioridad para los tratamientos a aplicar por el agente.
- ✓ Sería favorable que el SMA fuera capaz de recordar cuál es el tratamiento más efectivo en cada uno de los usuarios.

Bibliografía:

Chóliz, M. (2005). Psicología de la Emoción: el proceso emocional, Departamento de Psicología Básica. Universidad de Valencia: Pág. 1-33.

Gómez, D. D. (2009). Sistemas Multiagente para el diagnóstico presuntivo de enfermedades ginecológicas. Facultad de Informática, Universidad de Cienfuegos: Carlos Rafael Rodríguez: 146.

Jiménez, M. L. V. (2006). "Emociones Positivas." Papeles del Psicólogo**27**.

José A. Piqueras Rodríguez, et al. (2009). "EMOCIONES NEGATIVAS Y SU IMPACTO EN LA SALUD MENTAL Y FÍSICA." Suma Psicológica**16**.

Picard, R. (1997). "Affective computing", The MIT Press.

Pinto, R. P. (2006). EVALUACIÓN DE LA METODOLOGÍA PASSI EN UN CASO DE ESTUDIO FACULTAD DE INGENIERÍA INDUSTRIAL CENTRO DE ESTUDIOS DE INGENIERÍA DE SISTEMAS INSTITUTO SUPERIOR POLITÉCNICO "JOSÉ ANTONIO ECHEVERRÍA"

Rodríguez, L. R. (2008). "Enseñando Prolog con mapas conceptuales." Revista Cubana de Ciencias Informáticas**Vol.2**.

Rojas, L. M. H. (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. Departamento de Ingeniería Informática, Universidad "José Martí", Sancti-Spíritus: 60.

Sanz, J. J. G. (2003). "Metodologías para el desarrollo de sistemas multi-agente " Revista Iberoamericana de Inteligencia Artificial**18** 51-63.

Aguiar, E. (2006). Introduction SQLite.

Alzina, R. B. (2003). Educación Emocional.

Chóliz, M. (2005). Psicología de la Emoción: el proceso emocional, Departamento de Psicología Básica. Universidad de Valencia: Pág. 1-33.

Díaz, C. A. H. (2014). Red Neuronal Artificial para la estimación de la producción de biogás en el laboratorio del Grupo de Biogás de la UNISS.

Díaz, J. L. C. (2013). Sistema experto para el análisis y detección de enfermedades, Instituto Superior Minero Metalúrgico (Cuba).

Escofet, C. M. (2007). El lenguaje SQL.

Gómez, D. D. (2009). Sistemas Multiagente para el diagnóstico presuntivo de enfermedades ginecológicas. Facultad de Informática, Universidad de Cienfuegos: Carlos Rafael Rodríguez: 146.

Jiménez, M. L. V. (2006). "Emociones Positivas." Papeles del Psicólogo**27**.

José A. Piqueras Rodríguez, et al. (2009). "EMOCIONES NEGATIVAS Y SU IMPACTO EN LA SALUD MENTAL Y FÍSICA." Suma Psicológica**16**.

Picard, R. (1997). "Affective computing", The MIT Press.

Pinto, R. P. (2006). EVALUACIÓN DE LA METODOLOGÍA PASSI EN UN CASO DE ESTUDIO FACULTAD DE INGENIERÍA INDUSTRIAL CENTRO DE ESTUDIOS DE INGENIERÍA DE SISTEMAS INSTITUTO SUPERIOR POLITÉCNICO "JOSÉ ANTONIO ECHEVERRÍA"

Pinto, R. P. (2006). Evaluación de la Metodología PASSI en un caso de estudio., CUJAE: INSTITUTO SUPERIOR POLITÉCNICO "JOSÉ ANTONIO ECHEVERRÍA".

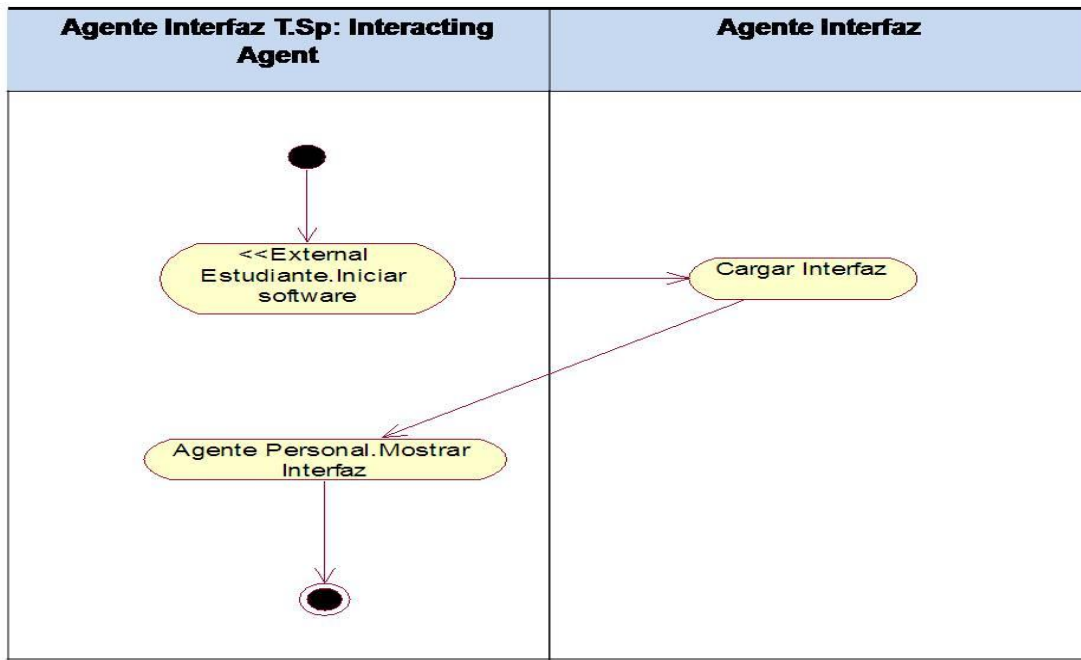
Rodríguez, L. R. (2008). "Enseñando Prolog con mapas conceptuales." Revista Cubana de Ciencias Informáticas**Vol.2**.

Rojas, L. H. (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. Departamento de Ingeniería Informática, Universidad "José Martí", Sancti-Spíritus: 60.

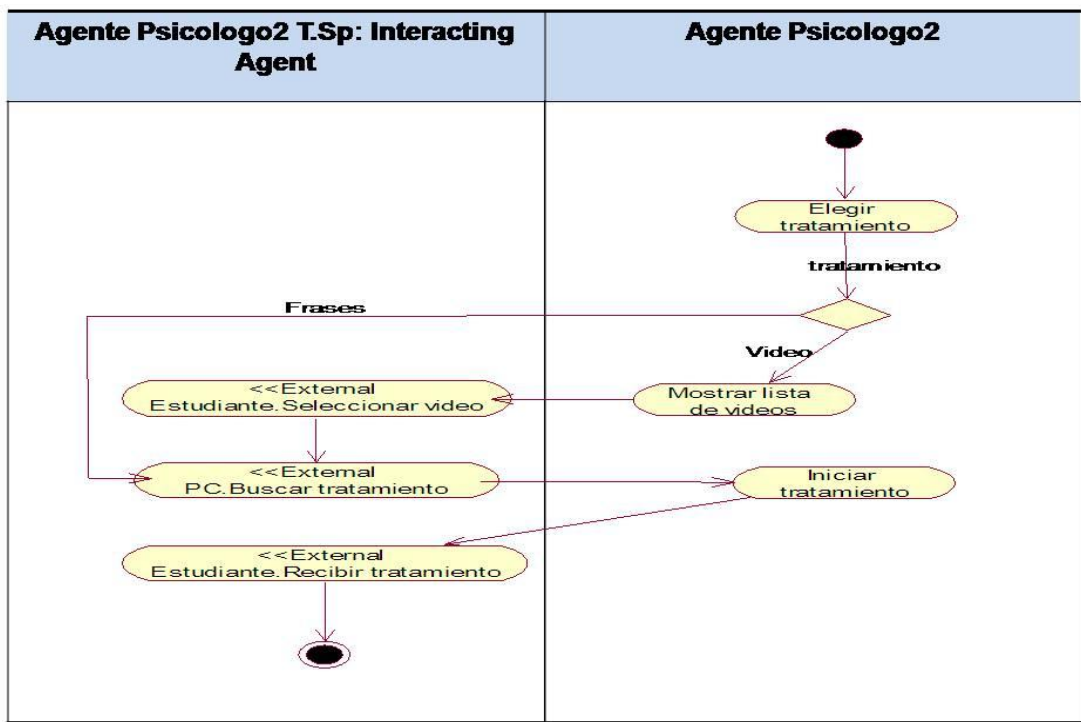
Sánchez, J. (2003). "JAVA 2." 255.

Sanz, J. J. G. (2003). "Metodologías para el desarrollo de sistemas multi-agente " Revista Iberoamericana de Inteligencia Artificial**18** 51-63.

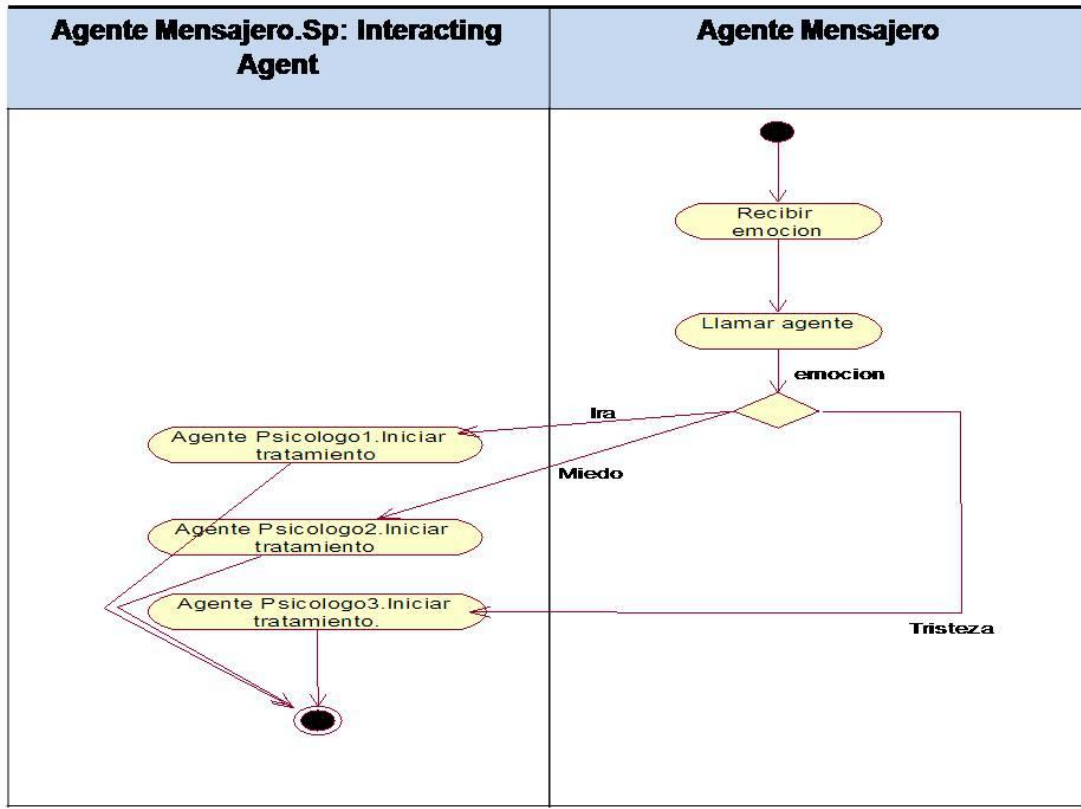
Anexos.



Anexo 1. Diagrama de actividad del agente interfaz



Anexo 2. Diagrama de actividad del agente psicólogo 2



Anexo 3. Diagrama de actividad del agente mensajero

Anexo 4: Biblioteca Reutilizable de código

-Agente Interfaz.

```
public class Agente Interfaz extends Agent {

    private static String AGENT_TYPE = Agente Interfaz;
    public String version = 1.0;
    private Usuario usuario ;
    private Emoción emocion ;
    private Modificar update ;
    private GetUser getUser ;
    private Create User createUser ;
    private Delete User delete ;
    private GetTratamiento getTratamiento ;
    private GetEmoción getEmocion ;
    private Tratar Ira tratarIra ;
    private Tratar Tristeza tratarTristeza ;
    private Tratar Miedo tratarMiedo ;
    private Validar validar ;

    public void Agente Interfaz(String platform, String name,
    String ownership) {
        //insert here your code
    }
}
```

```

    }

    public void shutdown() {
        //insert here your code
    }

    private void setup() {
        //insert here your code
    }

    private void register_WithAMS() {
        //insert here your code
    }

    private void register_WithDF() {
        //insert here your code
    }

```

Agente Mensajero:

```

public class Agente Mensajero extends Agent {

    private static String AGENT_TYPE = Agente Mensajero;
    public String version = 1.0;
    private Usuario usuario ;
    private Emoción emocion ;
    private Modificar upDate ;
    private GetUser getUser ;
    private Create User createUser ;
    private Delete User deleteUser ;
    private GetTratamiento getTratamiento ;
    private GetEmoción getEmocion ;
    private Tratar Ira tratarIra ;
    private Tratar Tristeza tratarTristeza ;
    private Tratar Miedo tratarMiedo ;
    private Validar validar ;

    public void Agente Mensajero(String platform, String name,
String ownership) {
        //insert here your code
    }

    public void shutdown() {
        //insert here your code
    }

    private void setup() {
        //insert here your code
    }

    private void register_WithAMS() {
        //insert here your code
    }

    private void register_WithDF() {

```



```
    //insert here your code
}
```

Agente Psicólogo 2:

```
private static String AGENT_TYPE = Agente Psicologo2;
public String version = 1.0;
private Emoción emocion ;
private Tratar Miedo miedo ;
private GetTratamiento getTratamiento ;
private GetEmoción getEmocion ;
private GetTratamientoM getTratamientoM ;

public void Agente Psicologo2(String platform, String name,
String ownership) {
    //insert here your code
}

public void shutdown() {
    //insert here your code
}

private void setup() {
    //insert here your code
}

private void register_WithAMS() {
    //insert here your code
}

private void register_WithDF() {
    //insert here your code
}
```

Agente Psicólogo3:

```
public class Agente Psicologo3 extends Agent {

private static String AGENT_TYPE = Agente Psicologo3;
public String version = 1.0;
private Emoción emocion ;
private Tratar Tristeza tristeza ;
private GetTratamiento getTratamiento ;
private GetEmoción getEmocion ;
private GetTratamientoT getTratamientoT ;

public void Agente Psicologo3(String platform, String name,
String ownership) {
    //insert here your code
}

public void shutdown() {
    //insert here your code
}

private void setup() {
```

```
    //insert here your code
}

private void register_WithAMS() {
    //insert here your code
}

private void register_WithDF() {
    //insert here your code
}
```