



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
La Universidad Católica de Loja

ÁREA TÉCNICA

TITULACIÓN DE INGENIERO EN SISTEMAS INFORMÁTICOS
Y COMPUTACIÓN

**Arquitectura multiagente para detectar estados emocionales de los
estudiantes en un curso virtual.**

TRABAJO DE FIN DE TITULACIÓN.

AUTOR: Cabrera Montoya, Karla Yolanda

DIRECTOR: Valdiviezo Díaz, Priscila Marisela, Mgs.

LOJA-ECUADOR

2014

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN

Magister.

Priscila Marisela Valdiviezo Díaz

DOCENTE DE LA TITULACIÓN

CERTIFICA:

Que el presente trabajo, denominado: "Arquitectura multiagente para detectar estados emocionales de los estudiantes en un curso virtual" realizado por el profesional en formación: Cabrera Montoya Karla Yolanda; cumple con los requisitos establecidos en las normas generales para la Graduación en la Universidad Técnica Particular de Loja, tanto en el aspecto de forma como de contenido, por lo cual me permito autorizar su presentación para los fines pertinentes.

Loja, marzo de 2014

f).

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Cabrera Montoya Karla Yolanda declaro ser autor (a) del presente trabajo de fin de titulación: Arquitectura multiagente para detectar estados emocionales de los estudiantes en un curso virtual, de la Titulación de Ingeniero en Sistemas Informáticos y Computación, siendo Priscila Marisela Valdiviezo Díaz director (a) del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”

f.....

Autor: Cabrera Montoya Karla Yolanda

Cédula: 1104592504

DEDICATORIA

El presente proyecto de tesis se lo dedico principalmente a Dios, por permitirme vivir esta maravillosa experiencia y hacer posible que este sueño se haga realidad.

A mis Padres Angel y Yolita, por el incondicional apoyo a lo largo de mi vida y en particular en mis estudios, por todos los valores que supieron inculcarme desde mi niñez los mismos que me han servido para lograr alcanzar todas las metas que me he trazado.

A mis hermanos, por alentarme y apoyarme siempre en los momentos difíciles.

A mis queridos sobrinos, para quienes deseo que esta experiencia les sirva de ejemplo, y luchen por cumplir sus objetivos.

A mi querido Fabricio, por brindarme todo su apoyo y comprensión, por alentarme siempre para seguir adelante y culminar con éxito mis estudios.

A los docentes que tuve a lo largo de mi vida estudiantil, ya que compartieron conmigo sus conocimientos, para poder culminar con mi carrera.

A mis compañeros y amigos que siempre han estado conmigo cuando los necesito, y a todas las personas que no creyeron en mí, porque su desconfianza me dio fuerzas para seguir adelante y no dejarme vencer como muchos de ellos.

AGRADECIMIENTO

Agradezco a Dios por darme la oportunidad y la sabiduría para obtener este logro alcanzado.

De manera especial agradezco a la Ing. Priscila Valdiviezo, por haberme guiado, exigido y motivado durante el desarrollo de mi tesis, infinitas gracias.

A mis queridos padres, a mi familia por todo el cariño y el apoyo brindado durante mi carrera.

También a mis amigos con quienes pude compartir esta gran experiencia de vida, para poder estar donde me encuentro ahora.

A todos gracias Totales...

ÍNDICE DE CONTENIDOS

CARÁTULA.....	i
APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN.....	ii
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDOS.....	vi
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE TABLAS.....	x
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
OBJETIVOS.....	5
CAPÍTULO 1: ESTADO DEL ARTE.....	6
1.1 Introducción.....	7
1.2 Análisis de Sentimientos en E-Learning.....	7
1.3 Clasificación de las Emociones.....	8
1.3.1 Emociones que interfieren en el aprendizaje: Ansiedad y Frustración.....	9
1.4 Clasificación de la Polaridad.....	9
1.5 Clasificación de la Intensidad.....	10
1.6 Modelos de Emociones.....	10
1.6.1 Modelo Circumplejo de Russell.....	11
1.6.2 Modelo de aprendizaje en espiral de Kort.....	11
1.7 Modelado de las Emociones: Emotion Markup Lenguaje.....	12
1.7.1 Elementos del Marcado de Emociones.....	14
1.8 Metodologías para la construcción de agentes.....	15
1.8.1 Comparación de Metodologías.....	16
1.9 Arquitecturas para construir Agentes.....	20
1.9.1 Deliberativas.....	20
1.9.2 Reactivas.....	22
1.9.3 Híbridas.....	24
1.9.4 Comparación de Arquitecturas.....	24
1.10 Arquitecturas Multiagente.....	27
1.11 Plataforma de desarrollo del Sistema Multiagente.....	31
1.11.1 JADE.....	31

1.12	Técnicas para la Extracción de Emociones.....	32
1.12.1	Extracción de Emociones a través de Texto.....	32
1.12.2	Extracción de Emociones a través de Texto e Imagen.....	36
1.12.3	Modelado de la Frustración.....	42
1.13	Herramientas.....	45
1.13.1	Herramientas para la extracción de emociones en texto.....	45
1.13.2	Herramientas para la extracción de emociones en video.....	49
1.14	Selección de Técnicas y Herramientas.....	52
1.15	Trabajos similares y Arquitecturas propuestas.....	53
CAPÍTULO 2: CARACTERÍSTICAS DE LOS ESTADOS AFECTIVOS QUE INFLUYEN EN EL APRENDIZAJE.....		58
2.1	Introducción.....	59
2.2	Análisis de Texto.....	59
2.2.1	Emoticones.....	59
2.2.2	Alteraciones Lingüísticas.....	60
2.2.3	Exageración de Signos de Puntuación.....	61
2.2.4	Onomatopeyas.....	62
2.2.5	Acrónimos.....	62
2.2.6	Diccionario Raíz.....	62
2.2.7	StopWord List.....	65
2.3	Análisis de los diálogos que presentan estados emocionales.....	66
2.3.1	Escenario: Extracción de palabras que determinan el tipo de emoción en texto, a través de la Red Social del EVA.....	67
2.3.2	Palabras que determinan el estado emocional.....	71
2.3.3	Características de los estados emocionales.....	72
CAPÍTULO 3: APLICACIÓN DE LA METODOLOGÍA SELECCIONADA PARA EL DESARROLLO DE AGENTES DE EXTRACCIÓN DE EMOCIONES.....		74
3.1	Metodología seleccionada MAS-CommonKADS.....	75
3.2	Aplicación de la Metodología.....	77
3.2.1	Conceptualización.....	77
3.2.2	Estructura de Agentes de extracción de emociones en Texto.....	85
3.2.3	Estructura de Agentes de extracción de emociones en Video.....	104
CAPÍTULO 4: PROPUESTA DE LA ARQUITECTURA DE AGENTES.....		118
4.1	Introducción.....	119
4.2	Modelo de Diseño.....	119
4.2.1	Esquema del Agente detección de Emociones en texto.....	119

4.2.2	Esquema del Agente detección de Emociones en video.....	120
4.3	Componentes.....	121
4.4	Descripción de componentes.....	122
4.5	Arquitectura Propuesta.....	123
4.5.1	Comunicación entre Agentes.....	124
4.6	Desarrollo de la Arquitectura detección de emociones en Texto.....	125
4.6.1	Lenguaje de Programación.....	125
4.6.2	Implementación de los Agentes.....	125
4.6.3	Desarrollo de la Arquitectura.....	127
	CONCLUSIONES.....	129
	RECOMENDACIONES Y TRABAJOS FUTUROS.....	130
	BIBLIOGRAFÍA.....	131
	ANEXOS.....	137

ÍNDICE DE FIGURAS

FIGURAS CAPÍTULO 1

Figura 1.1 Modelo Circumplejo de las Emociones.....	11
Figura 1.2 Modelo de Aprendizaje en Espiral.....	12
Figura 1.3 Interprete BDI.....	21
Figura 1.4 Arquitectura de Agentes Reactivos.....	22
Figura 1.5 Aprendizaje con un árbol de decisión.....	39
Figura 1.6 Clasificación con un árbol de decisión.....	40
Figura 1.7 Descomposición de una red neuronal.....	42
Figura 1.8 Resultados de Polaridad del Mensaje con AlchemyAPI.....	47
Figura 1.9 Resultados de Polaridad del Mensaje con SentiStrength.....	48
Figura 1.10 Resultados de Polaridad del Mensaje con NLTK.....	49
Figura 1.11 Resultados de Polaridad de la Imagen en Nvisio.....	51
Figura 1.12 Resultados de la Polaridad de la imagen en tiempo real con EmotionRecognition.....	52
Figura 1.13 Arquitectura de Agentes para el análisis de opinión.....	53
Figura 1.14 Arquitectura general del Agente.....	55
Figura 1.15 Arquitectura Multiagente para Sistemas Emergentes y Auto-Organizados...	57

FIGURAS CAPÍTULO 3

Figura 3.1 Modelos de MAS-CommonKADS.....	76
Figura 3.2 Casos de Uso para los Actores Activos.....	79
Figura 3.3 Descomposición de la Tarea T1: Análisis de Sentimientos.....	93
Figura 3.4 Casos de uso internos.....	94
Figura 3.5 Diagrama de secuencias para las interacciones de la conversación.....	101
Figura 3.6 Canales básicos de comunicación y flujo de conversaciones entre agentes	102
Figura 3.7 Interacciones genéricas de la conversación.....	102
Figura 3.8 Jerarquía de Clases.....	103
Figura 3.9 Descomposición de la Tarea T1: Análisis de Sentimientos Video.....	109
Figura 3.10 Casos de uso internos Análisis de Video.....	110
Figura 3.11 Diagrama de secuencias para las interacciones de la conversación de Video.....	115
Figura 3.12 Canales básicos de comunicación y flujo de conversaciones entre agentes de Video.....	116
Figura 3.13 Interacciones genéricas de la conversación de Video.....	116

FIGURAS CAPÍTULO 4

Figura 4.1 Esquema de Agente Simple detección de emociones en texto.....	120
Figura 4.2 Esquema de Agente Simple detección de emociones en video.....	121
Figura 4.3 Arquitectura propuesta detección de Emociones en Texto.....	123
Figura 4.4 Arquitectura propuesta detección de Emociones en Video.....	124
Figura 4.5 Diagrama de flujo Agente Extracción.....	125
Figura 4.6 Diagrama de flujo Agente Segmentación.....	126
Figura 4.7 Diagrama de flujo Agente Polaridad.....	126
Figura 4.8 Interfaz gráfica del RMA.....	127
Figura 4.9 Resultados del Sistema.....	128

ÍNDICE DE TABLAS

TABLAS CAPÍTULO 1

Tabla 1.1 Elementos de EmotionML.....	14
Tabla 1.2 Comparación de Metodologías.....	16
Tabla 1.3 Características Principales de las Arquitecturas.....	25
Tabla 1.4 Comparativa entre las Principales características de las Arquitecturas.....	25
Tabla 1.5 Comparación de herramientas para extracción de emociones en Texto.....	45
Tabla 1.6 Comparación de Herramientas para extracción de emociones en Video.....	50

TABLAS CAPÍTULO 2

Tabla 2.1 Emoticones en caracteres ASCII.....	59
Tabla 2.2 Palabras alineadas a emociones de tipo raíz.	63
Tabla2.3 StopWord List.....	65
Tabla2.4 Palabras obtenidas del análisis textual.....	72

TABLAS CAPÍTULO 3

Tabla 3.1 Identificación del Agente Extracción.....	85
Tabla 3.2 Identificación del Agente PLN.....	86
Tabla 3.3 Identificación del Agente Polaridad.....	86
Tabla 3.4 Identificación del Agente Ontología.....	87
Tabla 3.5 Identificación del Agente Organizador.....	87

Tabla 3.6 Identificación del Agente Interfaz.....	88
Tabla 3.7 Distribución tareas-agentes.....	89
Tabla 3.8 Identificación del Agente Extracción de Video.....	104
Tabla 3.9 Identificación del Agente Video.....	104
Tabla 3.10 Identificación del Agente Ontología de Video.....	105
Tabla 3.11 Identificación del Agente Interfaz de Video.....	105
Tabla 3.12 Distribución tareas-agentes de Video.....	106

RESUMEN

En el presente trabajo de fin de titulación se ha obtenido un modelo de Arquitectura Multiagente aplicando la metodología MAS-CommonKADS, y con ayuda del análisis de texto de los foros y comentarios de la Red Social del entorno virtual de aprendizaje EVA, de los estudiantes, se ha podido establecer la polaridad emocional de forma no intrusiva que presenta el texto mediante la utilización de técnicas de procesamiento de lenguaje natural, permitiendo así, dar a conocer al docente cuando es necesario proporcionar una retroalimentación a los estudiantes en sus respectivos componentes académicos.

PALABRAS CLAVES: análisis de sentimientos, mas_commonkads, emociones y aprendizaje, arquitectura multiagente, agentes inteligentes, sistemas informáticos, minería de opinión, análisis contenido.

ABSTRACT

In this research it is proposed a multiagent architecture model based on MAS-CommonKADS methodology. Complemented by analysis of forum and social network, comments of the student it has been possible to establish emotional polarity of a non-intrusive way, representing text through the use of natural language process techniques, allowing to advise to the teacher when it is necessary to give feedback to students.

KEYWORDS: sentiment analysis, mas_commonkads, emotions and learning, multi-agent architecture, intelligent agents, computer systems, opinion mining, content analysis.

INTRODUCCIÓN

Ante la gran cantidad de texto con contenido subjetivo, surge el análisis de sentimientos que trata de clasificar el texto según su polaridad, mediante la utilización de técnicas de PLN y minería de texto. Al ser un tema de actualidad tiene mucho potencial para ser explotado al máximo, se lo puede trabajar desde varios puntos, ya sea el desarrollo de aplicaciones de escritorio, web o móviles, o como se lo maneja actualmente en algunas redes sociales.

Por estas razones se cree conveniente el desarrollo de una Arquitectura Multiagente para detectar estados emocionales, en el ámbito estudiantil especialmente en educación a distancia, ya que al no existir una interacción “cara a cara” entre el docente y el estudiante se dificulta su comunicación y el docente no puede brindar una retroalimentación necesaria de los temas con los cuales los estudiantes tiene ciertos problemas de aprendizaje.

Este trabajo de fin de titulación está compuesto por cuatro capítulos, detallados a continuación:

En el capítulo 1, se exploran los conceptos básicos de emociones relacionadas con el aprendizaje, así como las técnicas empleadas para la detección de emociones tanto en texto como en video, además de un estudio breve de las metodologías empleadas para el desarrollo de la arquitectura propuesta.

En el capítulo 2, se cree conveniente el planteamiento de escenarios, con la finalidad de comprender ciertos aspectos que debemos considerar dentro del análisis de texto, como son ciertas alteraciones en la lingüística, con la utilización de los escenarios se obtiene ciertas características de los estados afectivos que influyen en el aprendizaje de los estudiantes a partir de la utilización de stopword List, diccionario raíces y conceptos básicos de las alteraciones de la lingüística.

En el capítulo 3, se aplica la metodología seleccionada para el desarrollo de la arquitectura, en este caso se utilizó la metodología MAS_CommonKADS, que está orientada al desarrollo de sistemas multiagente mediante el uso de modelos para la realización de la conceptualización y análisis, mientras que su implementación se realizó independientemente de estos modelos mencionados, ya que esta metodología nos ofrece la funcionalidad de trabajar los requerimientos independientemente del diseño o implementación.

Finalmente, en el capítulo 4, se presenta el modelo de diseño de la Arquitectura multiagente propuesta, un modelo de agentes simple, así como los componentes que conforman dicha

arquitectura, su implementación en JADE y sus pruebas correspondientes. Y finalmente las conclusiones y recomendaciones a las cuales hemos llegado luego de realizar el trabajo de tesis.

OBJETIVOS

Objetivo General:

- ✓ Explorar las diferentes posibilidades destinadas a la extracción automática de las emociones de los estudiantes que interactúan en el EVA.

Objetivos Específicos:

- ✓ Conocer las diferentes emociones que están presentes en el aprendizaje online.
- ✓ Plantear escenarios para determinar las características de los estados afectivos que influyen en el aprendizaje.
- ✓ Investigar las diferentes metodologías, técnicas y herramientas que permitan desarrollar la Arquitectura multiagente para la extracción de emociones.
- ✓ Proponer una arquitectura multiagente para la extracción no intrusiva de emociones de los estudiantes que interactúan en el EVA.

CAPÍTULO 1
ESTADO DEL ARTE

1.1 Introducción

Ante el gran crecimiento de las TIC's en la Web 2.0, se han desarrollado varios proyectos orientados al análisis de datos en varios ámbitos, entre ellos el Análisis de Sentimientos o Minería de Opinión, que hacen referencia a la aplicación de técnicas de procesamiento de lenguaje natural (PLN), minería de textos, lingüística computacional, etc... utilizados para la extracción de información subjetiva generada por los usuarios (foros, redes sociales, recursos, tareas, mensajería, cuestionarios, etc.), para poder determinar ciertos patrones de comportamiento.

La minería de opinión trata de clasificar la información de acuerdo a lo que expresa el usuario, entre sus aplicaciones tenemos la determinación de la polaridad de opiniones (positivas / negativas), cabe recalcar que toda minería de datos tiene un margen de error, pero si es realizada correctamente este margen de error tiende a desaparecer.

En este capítulo se investiga los conceptos básicos relacionados con los estados emocionales de las personas, así como algunas técnicas utilizadas para la extracción de emociones.

1.2 Análisis de sentimientos en E-Learning

La incorporación de las TIC como recursos para el aprendizaje universitario supone uno de los retos actuales para la renovación de metodologías y estrategias didácticas. Sin embargo, por su parte otras investigaciones científicas documentan una tasa de abandono importante en los sistemas de teleformación y E-learning, fenómeno en el que parece tener un importante papel la motivación y la emoción del alumnado así como las características de los entornos virtuales para la formación (Rebollo M. , García, Barragán, Buzón, & Vega, 2008).

Las emociones sirven como un potente vehículo para aumentar o inhibir el aprendizaje (Greenleaf, 2003). Las emociones están vinculadas directamente a las actividades de logros o resultados de rendimiento. Por ejemplo, la satisfacción surge del aprendizaje, el aburrimiento con la rutina de la enseñanza en clase, o frustración e ira cuando se trata de tareas difíciles (Pekrun, 2006).

El Análisis de sentimientos es una característica de dos dimensiones de un texto afectado por dos factores primarios: la objetividad/ subjetividad del texto y el nivel de emoción positiva

o negativa inherente en el texto. Estos dos rangos juntos descritos por (Fakhraie, 2012) componen las dos dimensiones de los sentimientos de un texto:

Objetividad/ subjetividad: El nivel de conocimiento de los hechos reportados se indica mediante esta variable. Una nota objetiva es donde se hace referencia a los hechos verificables, mientras que, una nota subjetiva se basa principalmente en los sentimientos y pensamientos del escritor.

Emoción Positiva/Negativa: Esta variable nos permite conocer el nivel de emociones positivas y negativas de una nota.

1.3 Clasificación de las emociones

Para poder determinar la clasificación de las emociones, primero debemos conocer el concepto de emociones. (Oatley, 1992) Manifiesta que una emoción puede definirse como *“una experiencia afectiva en cierta medida agradable o desagradable, que supone una cualidad fenomenológica característica y que compromete tres sistemas de respuesta: cognitivo-subjetivo, conductual-expresivo y fisiológico-adaptativo”*.

(Goleman, 1996), manifiesta que el término emoción hace referencia a “los sentimientos y pensamientos, los estados biológicos, los estados psicológicos y el tipo de tendencia a la acción que lo caracteriza”.

Así mismo, (Goleman, 1996) clasificó las emociones de la siguiente manera:

- ✓ **Ira:** rabia, enojo, resentimiento, furia, exasperación, indignación, acritud, animosidad, irritabilidad, hostilidad, y en casos más extremos, odio y violencia.
- ✓ **Tristeza:** aflicción, pena, desconsuelo, pesimismo, melancolía, autocompasión, soledad, desaliento, desesperación, y en casos patológicos, depresión grave.
- ✓ **Miedo:** ansiedad, aprensión, temor, preocupación, consternación, inquietud, desasosiego, incertidumbre, nerviosismo, angustia, susto, terror, y en caso de ser psicopatológico, fobia y pánico.
- ✓ **Alegría:** felicidad, gozo, tranquilidad, contento, beatitud, deleite, diversión, dignidad, gratificación, satisfacción, euforia, capricho, éxtasis y en un caso extremo, manía.
- ✓ **Amor:** aceptación, cordialidad, confianza, amabilidad, afinidad, devoción, adoración, enamoramiento.
- ✓ **Sorpresa:** sobresalto, asombro, desconcierto, admiración.
- ✓ **Aversión:** desprecio, desdén, displicencia, asco, antipatía, disgusto y repugnancia.
- ✓ **Vergüenza:** culpa, perplejidad, desazón, remordimiento, humillación, pesar, aflicción.

En el estudio realizado por (Blanco & Carracedo, 2009) nos presentan otro tipo de clasificación de las emociones:

- ✓ **Emociones positivas y negativas**, consiste en agrupar las emociones y jerarquizarlas, determinando así su polaridad (positiva/negativa). Las negativas son las emociones que no nos agradan o que nos perjudican y positivas aquellas que nos agradan o nos benefician.
- ✓ **Los tres ejes**, aquí se sitúa el sistema de elección de Joel Davittz y Klaus Scherer, que desarrollaron un método para evaluar que emociones deben ser consideradas como básicas, dicho método utiliza los tres ejes o dimensiones de un campo semántico (se relacionaban con los distintos aspectos de la voz).
 - **Potencia o fuerza**, corresponde a la atención/rechazo, es decir, el ritmo, la inflexión y la enunciación.
 - **Valencia, agrado o valoración**, según lo placentero o desagradable de la emoción.
 - **Actividad**, hace referencia a la velocidad, timbre y tono de voz.

1.3.1 Emociones que interfieren en el aprendizaje: Ansiedad y Frustración

Frustración, se produce cuando algo o alguien impide que un individuo avance o culmine con éxito una meta en particular. La frustración no es diferente a una respuesta negativa afectiva común a una variedad de situaciones (ansiedad). Ansiedad es más que una respuesta emocional, también consiste en respuestas conductuales, cognoscitivas, y fisiológicas (Seligman, Walker, & Rosenhan, 2001).

1.4 Clasificación de la polaridad

Con esta clasificación se pretende determinar la polaridad de un texto (clasificarlo en positivo/negativo) de acuerdo a su carga emocional. La clasificación de la polaridad ha sido abordada desde dos enfoques diferentes, métodos supervisados y no supervisados. En los métodos supervisados se trata a la clasificación de la polaridad como un problema de clasificación binaria (positivo vs. negativo), donde un conjunto de datos etiquetados previamente es usado como un conjunto de entrenamiento para el algoritmo de aprendizaje máquina, mientras que, los métodos no supervisados se apoyan en la idea de las denominadas expresiones polares (son palabras o conjunto de palabras que poseen cierta polaridad en sí mismas) para determinar la polaridad total de un texto. Para poder realizar el

cálculo global de la polaridad del texto en función a expresiones polares, se utiliza fórmulas matemáticas, como la métrica PMI (Pointwise mutual information), que se basa en la frecuencia de co-ocurrencia de los términos y se calcula en base a la siguiente ecuación (Carrillo, 2011):

$$PMI(term_1, term_2) = \log_2 \left(\frac{\Pr(term_1 \wedge term_2)}{\Pr(term_1) * \Pr(term_2)} \right) \quad (1)$$

(Pang, Lee, & Vaithyanathan, 2002), fueron los pioneros en utilizar métodos supervisados para la clasificación de texto en positivo y negativo observando su polaridad. Además plantearon una hipótesis de “si los métodos utilizados para la tarea de clasificación de tópicos son aplicables a la detección de polaridad en críticas de películas, o por el contrario, es preciso desarrollar aproximaciones específicas”, para comprobar dicha hipótesis realizaron varios experimentos con tres algoritmos de aprendizaje máquina: Naïve Bayes, Maximum Entropy y Support Vector Machines, cuyos atributos fueron la frecuencia de los términos, su presencia, el uso de adjetivos, n-gramas y categorías gramaticales de los términos. A la conclusión que obtuvieron es que el uso de unigramas, junto con los términos y el algoritmo Support Vector Machines permiten la obtención de mejores resultados al momento de clasificar, por ejemplo críticas de cine.

1.5 Clasificación de la Intensidad

Es la tarea del análisis de sentimientos que aborda el problema de clasificar un texto en diferentes clases de intensidad de acuerdo a su carga emocional. Se puede diferenciar entre las aproximaciones que abordan el problema de clasificar los textos en tres tipos de intensidad (positiva, neutra y negativa) y las aproximaciones que van un paso más allá y abordan el problema de clasificar los textos en más de tres clases (fuerte positivo, positivo, neutral, negativo y fuerte negativo) (Carrillo, 2011).

1.6 Modelos de emociones

El objetivo de este estudio es comprender cómo las emociones de los alumnos evolucionan durante el proceso de aprendizaje, con el fin de desarrollar sistemas de aprendizaje que

reconozcan y respondan apropiadamente a su cambio emocional. Existen varios autores que clasifican las emociones en base a modelos, entre ellos tenemos:

- ✓ Modelo Circumplejo de Russell
- ✓ Modelo de aprendizaje en espiral de Kort

1.6.1 Modelo circumplejo de Russell

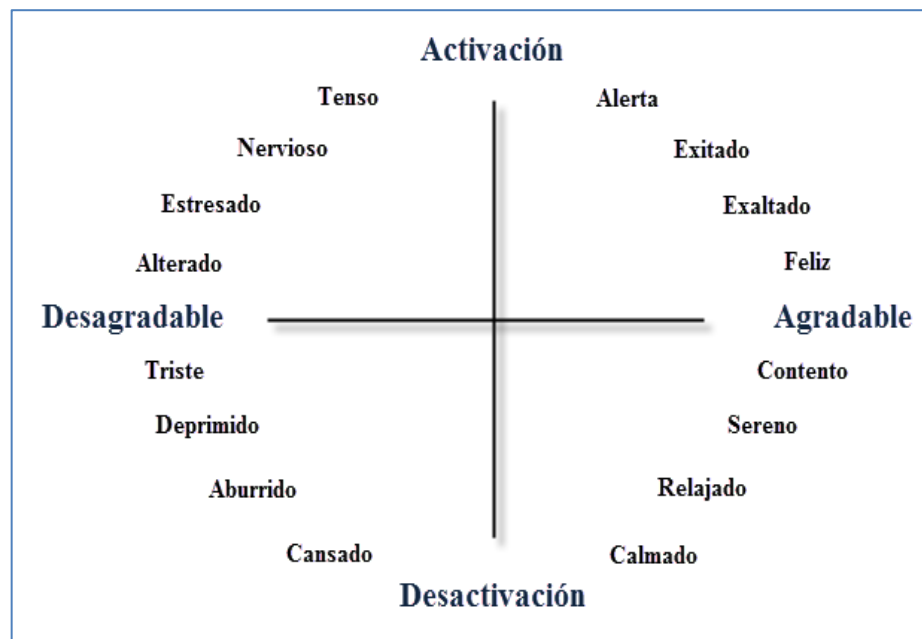


FIGURA 1.1 modelo circumplejo de las emociones

Fuente: (Barrett & Russell, 1999).

Según (Barrett & Russell, 1999) en su modelo propuesto describen las emociones como un espacio bipolar cuyas dimensiones son independientes, en una de ellas se obtiene el valor emocional positivo o negativo, mientras que en la otra refleja el nivel de activación alta o baja. Las dimensiones a pesar de ser independientes, ambas son necesarias para comprender claramente las emociones.

En el modelo de Russell las emociones se ven como combinaciones de excitación y valencia; están distribuidas en un sistema de coordenadas donde el eje Y indica el grado de excitación o activación y el eje X mide la valencia desde las emociones negativas a las positivas (Akbiyik, 2010).

1.6.2 Modelo de aprendizaje en espiral de Kort

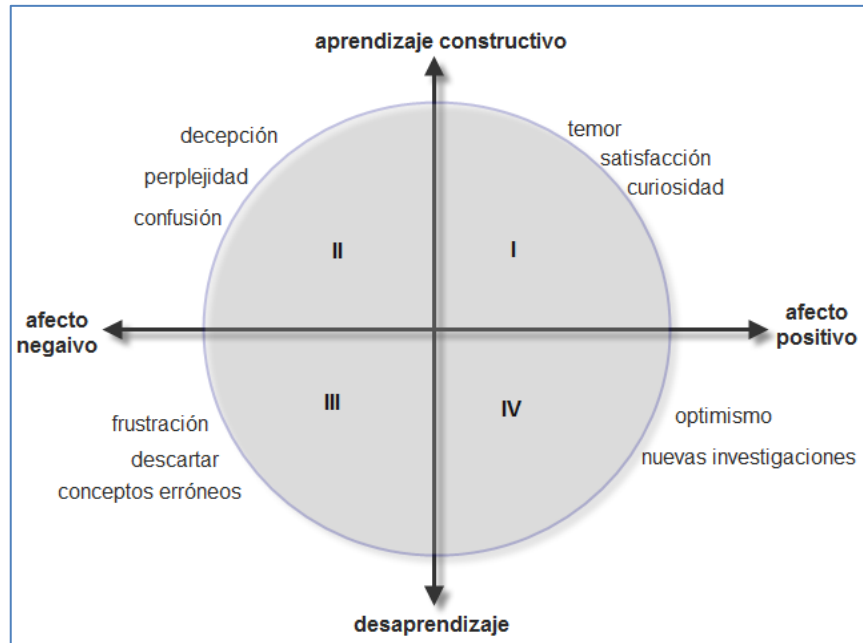


FIGURA 1.2 modelo de aprendizaje en espiral

Fuente:(Kort, Reilly, & Picard, 2001)

En el modelo en espiral propuesto por (Kort, Reilly, & Picard, 2001) de los cuatro cuadrantes del aprendizaje, las emociones cambian según el movimiento del estudiante en los cuadrantes y en la espiral, también propusieron cinco conjuntos de emociones que pueden ser relevantes para el aprendizaje. No obstante, se necesita de evidencia empírica para validar el modelo de espiral del aprendizaje y confirmar los efectos que estas emociones pueden tener sobre el aprendizaje.

1.7 Modelado de las emociones: Emotion Markup Language

Los sistemas computarizados necesitan un formato de representación, para poder reconocer, simular y procesar información relacionada con las emociones. Para conseguir una mejor interoperabilidad es necesario utilizar un formato de representación estándar(Schröder, y otros, 2011).

Como cualquier formato estándar, su objetivo principal es doble: permitir que un componente tecnológico represente y procese datos, y permitir la interoperabilidad entre los diferentes componentes tecnológicos de procesamiento de datos. Los casos de uso de EmotionML se puede agrupar en tres tipos generales(Baggia, Burkhardt, Pelachaud, Peter, & Zovato, 2012):

- Anotación manual de materiales que incluyan emotividad, tales como: videos, grabación de: voz, rostros, y textos, etc.
- Reconocimiento automático de emociones desde sensores, incluyendo sensores fisiológicos, grabaciones de voz, expresiones faciales, etc., así como la combinación de sensores multimodales.
- Generación de respuestas de los sistemas relacionados con las emociones, que pueden implicar el razonamiento sobre las implicaciones emocionales de los eventos, la prosodia emocional en voz sintética, expresiones faciales y gestos de los agentes incorporados o robots, la selección de música y los colores de iluminación en una habitación, etc.

(Baggia, Burkhardt, Pelachaud, Peter, & Zovato, 2012), proporcionan ejemplos concretos de las tecnologías existentes en las que se podría aplicar EmotionML, e incluyen:

- ✓ **Minería de opinión/Análisis de sentimiento en la Web 2.0**, para la realización de seguimientos automáticos sobre la actitud de los clientes respecto a un producto a través de blogs.
- ✓ **Monitoreo afectivo**, como aplicaciones ambientales de vida asistida para la detección de personas de edad avanzada, detección de miedo con fines de vigilancia, o el uso de sensores portátiles para comprobar la satisfacción del cliente.
- ✓ Diseño de personajes y el control de los juegos y mundos virtuales.
- ✓ **Robots sociales**, tales como robots guía acoplados con los visitantes.
- ✓ **Síntesis vocal expresiva**, generando voz sintética con diferentes emociones (feliz/triste, amable/apologético); por ejemplo: poner más información a disposición de personas con discapacidad visual y enriquecerlos con sus contenidos.
- ✓ **Reconocimiento emocional**, por ejemplo: para detectar clientes enojados.
- ✓ **Apoyo a las personas con discapacidad**, tal como programas educacionales para personas con autismo. EmotionML puede ser utilizado para hacer la intensión emocional del contenido explícito. Esto permitiría a las personas con dificultad de aprendizaje (tal como síndrome de Asperger) realizar el contexto emocional del contenido.
- ✓ EmotionML puede ser usado para **transcripciones de medios de comunicación y subtítulos**. Cuando las emociones están marcadas para ayudar a personas con dificultades auditivas que no pueden escuchar la banda sonora, se pone más información a disposición para enriquecer su experiencia en el sitio.

1.7.1 Elementos del marcado de emociones

Tabla 1.1 elementos de EmotionML

Anotación	<emotionml>
Definición	El elemento raíz de un documento EmotionML.
Hijo	El elemento puede contener uno o más elementos <emotion>. Puede contener un solo elemento <info>. Puede contener uno o más <vocabulary>
Atributos	<ul style="list-style-type: none"> • Requerido: <ul style="list-style-type: none"> ○ Declaración de espacio de nombres para EmotionML ("http://www.w3.org/2009/10/emotionml") ○ version indica la versión de la especificación que se utilizará para el documento. Los documentos que utilizan esta especificación deberá utilizar 1,0 para el valor • Opcional: <ul style="list-style-type: none"> ○ category-set declara un vocabulario categoría mundial. El atributo debe ser de tipo xsd: anyURI y debe hacer referencia a la identificación de un <vocabulary> elemento que define un vocabulario de emoción con type = "category". ○ dimension-set declara un vocabulario dimensión global. El atributo debe ser de tipo xsd: anyURI y debe hacer referencia a la identificación de un <vocabulary> elemento que define un vocabulario de emoción con type= "dimension". ○ appraisal-set declara un vocabulario de evaluación global. El atributo debe ser de tipo xsd: anyURI y debe hacer referencia a la identificación de un <vocabulary> elemento que define un vocabulario de emoción con type = "appraisal". ○ action-tendency-set declara un vocabulario de acción de tendencia global. El atributo debe ser de tipo xsd: anyURI y debe hacer referencia a la identificación de un <vocabulary> elemento que define un vocabulario de emoción con type = "action-tendency".
Aparición	Este es el elemento raíz, no puede ocurrir tal como los hijos en ningún otro elemento de EmotionML

Fuente:(Baggia, Burkhardt, Pelachaud, Peter, & Zovato, 2012)

1.8 Metodologías para la construcción de agentes

Son técnicas de análisis y diseño de sistemas de agentes, que surgen mediante extensiones de las metodologías existentes orientadas a objetos y de ingeniería del conocimiento, que permiten la generalización en el desarrollo de aplicaciones orientadas a agentes (Iglesias, Garijo, & González, 1999).

Entre las Metodologías para la construcción de Agentes tenemos las siguientes:

- BDI (Creencias, Deseos, Intenciones) (Bratman, 1987)
 - Define dos vistas para modelar, la vista externa que consiste en descomponer el sistema en agentes y definir sus interacciones a través de los modelos de agentes e interpretación. La vista interna modela las clases de agentes mediante tres modelos: de creencia, objetivos y planificación.

- GAIA (Wooldridge, Jennings, & Kinny, 2000)
 - Esta metodología se aplica más para el mundo real, los agentes son sistemas de cómputo que utilizan importantes recursos, son heterogéneos. La especificación de requerimientos es completamente independiente del proceso de análisis y diseño.

- MAS_COMMONKADS (Iglesias, 1998)
 - Se basa en un modelo de experiencia, para desarrollar sistemas expertos que interactúan con el usuario considerando dos agentes básicos: el sistema y el usuario.

- MASE (DeLoach, 2011)
 - Es una metodología completa basada en el ciclo de vida clásico del software, con un entorno propio de desarrollo para analizar, diseñar y construir sistemas multiagentes heterogéneos.

- MESSAGE/UML (Message, 2000)
 - Se basa en una serie de definiciones de conceptos que añade a UML, propone el análisis y diseño del sistema de agentes definiendo 5 meta-modelos: Agente, Organización, Dominio, Tareas – objetivos, Interacciones.

- ZEUS (Nwana, Ndumu, Lee, & Collis, 1999)
 - Proporciona una plataforma de ejecución de agentes, prototipos de agentes, y componentes para su realización.

1.8.1 Comparación de metodologías

Para poder determinar que Metodología debemos utilizar para la creación del SMA, es necesario realizar una comparación de funcionalidades entre ellas, cuyo resultado se observa en la Tabla 1.2.

Tabla 1.2 comparación de metodologías

Metodología	Comparativa
BDI	<p>Ventajas:</p> <ul style="list-style-type: none"> ✓ Basada en el modelo cognitivo del ser humano ✓ Posee dos puntos de vista: interno (funcionamiento interno individual de cada uno de los agentes) y externo. <ul style="list-style-type: none"> ○ Interno <ul style="list-style-type: none"> ▪ Los agentes utilizan una representación interna del entorno y un modelo del mundo que los rodea. ▪ Son bastante versátiles a la hora de definir el entorno, objetivos y acciones que se pueden asignar al agente. ▪ El agente toma decisiones para interactuar con el mundo basadas en sus <i>Creencias</i> y guiado por sus <i>Deseos</i> (objetivos). ○ Externo <ul style="list-style-type: none"> ▪ Permiten detallar la jerarquía, instancias y otros detalles del agente. ▪ Identifica roles del dominio de la aplicación, así como las responsabilidades de cada rol y los servicios que provee. ▪ Identificar las interacciones asociadas con la provisión de cada servicio (eventos, acciones y performativas requeridas por cada

	<p>servicio).</p> <ul style="list-style-type: none"> ▪ Los roles son un conjunto de responsabilidades, y estas a su vez son conjuntos de servicios. <ul style="list-style-type: none"> ✓ Su objetivo principal son los roles, responsabilidades, servicios y metas de los agentes. ✓ Analiza las aplicaciones en términos de qué es aquello que debe ser conseguido y en qué contexto. ✓ Los objetivos comparados con los comportamientos o planes, son más estables en el dominio de la aplicación. Esto permite que el sistema sea más robustos, modular y estable. ✓ Se puede añadir nuevos planes para nuevos contextos son alterar los existentes (desarrollo incremental). <p>Inconvenientes:</p> <ul style="list-style-type: none"> ✓ Estos modelos no son suficientes por si solos, necesitan una arquitectura que los agrupe adecuadamente.
<p style="text-align: center;">GAIA</p>	<p>Ventajas:</p> <ul style="list-style-type: none"> ✓ Los agentes son heterogéneos (pueden implementarse en diferentes lenguajes, tecnologías, arquitecturas) ✓ Las agentes son sistemas de cómputo de grano grueso (uso importante de recursos) ✓ La organización de la estructura del sistemas (relaciones entre agentes), habilidades y servicios proveídos por los agentes son estáticos (no varían en el tiempo) ✓ La especificación de requerimientos es independiente del proceso de análisis y diseño del sistema ✓ Permite centrarse en la fase de análisis y diseño por separado ✓ Basada en roles de interacción. ✓ Diseño basado en agrupación de roles en agentes. <p>Inconvenientes:</p> <ul style="list-style-type: none"> ✓ Nivel de abstracción poco detallado del proceso a seguir.

MAS_COMMON KADS

Ventajas:

- ✓ Es una extensión de la metodología CommonKADS de desarrollo de sistemas expertos con:
 - Técnicas de orientación a objetos (OMT y OOSE)
 - Técnicas de ingeniería de protocolos: SDL y MSC
- ✓ Modelo de ciclo de vida en espiral dirigido por riesgos
 - Modelo en cascada con reutilización para proyectos pequeños
- ✓ Basado el desarrollo de un SMA en un conjunto de plantillas de modelos interrelacionados
- ✓ Las fases del proceso de desarrollo son:
 - Conceptualización
 - Definición de casos de uso
 - Análisis
 - Especificación de requerimientos del sistema a partir del desarrollo de modelos (excepto el diseño), guiado por los riesgos
 - Modelado de agentes
 - Estrategias y combinaciones de las mismas para identificar y desarrollar los agentes.

Inconvenientes:

- ✓ La metodología es muy compleja y amplia
- ✓ Demasiada información relevante que está en diversos sitios, dificultando su acceso y comprensión

MASE

Ventajas:

- ✓ Basada en el ciclo de vida clásico del software
- ✓ Analiza, diseña y construye sistemas multiagentes heterogéneos
- ✓ Los agentes son simples procesos que se comunican para conseguir el objetivo global del sistema
- ✓ Los agentes son vistos únicamente como una abstracción conveniente que puede o no tener inteligencia
- ✓ El comportamiento de los agentes está definido por autómatas.

MESSAGE/UML

- ✓ Soportado por la herramienta agentTool.
- ✓ Basado en RUP, trata especialmente las actividades de análisis y diseño.
- ✓ Ingeniar sistemas multiagentes prácticos
- ✓ Se divide en dos fases
 - Análisis: capturar objetivos, definir casos de uso y roles.
 - Diseño: creación y ensamblaje de clases de agentes, así como el diseño de comedia.

Inconvenientes:

- ✓ Está diseñada para modelar entornos con agentes estáticos no móviles

Ventajas:

- ✓ Se extiende de la ingeniería de software OO con conceptos del área de agentes
- ✓ Propone el análisis y diseño del sistema de agentes
- ✓ Se definen en 5 meta-modelos:
 - Agente: descripción detallada y extensa de cada agente y el rol dentro del sistema
 - Organización: captura la estructura global del sistema
 - Dominio: actúa como repositorio de información (para entidades y relaciones), concernientes al dominio del problema
 - Tareas-objetivos: determina qué hace el sistema de agentes, que objetivos persigue, y tareas implicadas
 - Interacciones: trata las interacciones a distintos niveles de abstracción
- ✓ El modelo de análisis se limita a generar modelos a partir de los meta-modelos
- ✓ Utiliza conceptos UML para modelar entidades
 - Objetos con atributos y operaciones realizadas por métodos
 - Desde el punto de vista de comportamiento son máquinas de estado

Inconvenientes:

- ✓ Diseño preciso y muy detallado, directamente se puede empezar a programar

ZEUS

Ventajas:

- ✓ Entorno virtual de desarrollo de SMA
- ✓ Proporciona una plataforma de ejecución de agentes, prototipos de agentes, y componentes para su realización.
- ✓ Se puede configurar agentes genéricos mediante la definición de:
 - Ontologías
 - Agentes
 - Tareas
 - Organización
 - Coordinación
- ✓ La etapa de desarrollo permite modelar iterativamente el sistema, puesto que se basa en análisis, diseño, implementación y soporte.
- ✓ En la etapa de realización de agentes se define previamente una ontología para enmarcar los agentes y su naturaleza o clase.

Inconvenientes:

- ✓ Diseño complejo, tiene que suministrar ontologías, reglas de comportamiento, planes de ejecución de tareas y mensajes a enviar a otros agentes

Fuente:(Bratman, 1987), (Wooldridge, Jennings, & Kinny, 2000), (Iglesias, 1998), (DeLoach, 2011), (Message,2000), (Nwana, Ndumu, Lee, & Collis, 1999)

1.9 Arquitecturas para construir Agentes

1.9.1 Deliberativas

(Maes, 1990), manifiesta que este tipo de arquitecturas utilizan modelos de representación simbólica del conocimiento, basadas en la teoría clásica de planificación, partiendo de un estado inicial para poder generar planes de consecución de objetivos.

Un agente deliberativo contiene un modelo simbólico del mundo, claramente representado, en el cual las decisiones se toman usando mecanismos de razonamiento lógico basados en la correspondencia de patrones y manipulación simbólica, para alcanzar los objetivos del agente (Corchado, 2005).

Según manifiesta (Corchado, 2005), para poder implementar una arquitectura deliberativa, primero debemos buscar una descripción simbólica apropiada del problema para luego integrarla en el agente, y este a su vez pueda razonar y realizar las tareas encargadas en un tiempo ya establecido.

- ✓ **Arquitectura BDI (Belief, Desires and Intentions – Creencias, Deseos e Intenciones)**, en este tipo de arquitecturas la toma de decisiones se realiza sobre un proceso de razonamiento que parte de las creencias que el agente tiene del mundo y considerando las intenciones y acciones. Esta arquitectura se fundamenta en la tradición filosófica el entendimiento del razonamiento práctico (deliberar y razonamiento de fines y medios “means-ends reasoning”), que es el proceso de decidir a cada momento que acción ejecutar para el cumplimiento de objetivos ya fijados. Los componentes básicos son: las creencias, deseos e intenciones del agente; las funciones que representan su deliberación y el razonamiento de fines y medios. Durante el proceso de decisión se trata de entender qué opciones están disponibles, cuando genera el grupo de alternativas se elige una de ellas y compromete (“commit”) con una, al momento que escogemos una opción se convierte en una intención, la cual nos permite determinar las acciones que va a realizar el agente. Las intenciones localizan el razonamiento práctico futuro del agente, cuando se tiene una intención en particular se descartan todas aquellas opciones que sean inconsistentes con la intención; cuando una intención es adoptada, el agente debe perseverar (“persist”) en ésta, pero cuando la intención ha cambiado o a su vez el agente sabe que no puede cumplir con ella puede rectificar. Existe una estrecha relación entre las intenciones y las creencias futuras, cuando el agente tiene una intención debe creer que tiene una gran posibilidad de cumplir con ella (Rizo, Llorens, & Pujol, 2003).

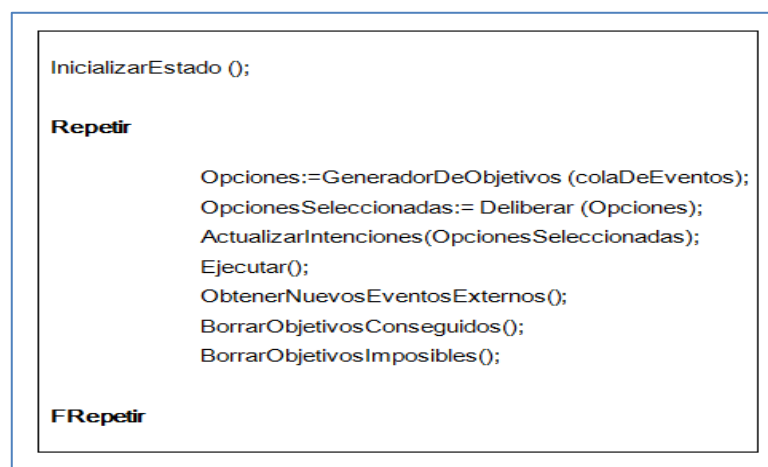


FIGURA 1.3interprete BDI
Fuente:(Rizo, Llorens, & Pujol, 2003).

El principal problema en el diseño del razonamiento práctico de los agentes es el adquirir un buen equilibrio entre los diferentes intereses. En algunas ocasiones los agentes deben abandonar sus intenciones (inalcanzables, logradas, la razón por la cual se originó la intención ya no existe), lo cual provoca que el agente reconsidere sus intenciones en ciertos puntos, produciendo un dilema al momento de crear un equilibrio entre comportamiento pro-activo (intencionado) y reactivo (condicionado a eventos). La dinámica del entorno condiciona este equilibrio, mientras más dinámico, la habilidad de reacción a cambios modificando las intenciones se vuelve más importante. Entre las razones que hacen más interesante este modelo tenemos: por ser intuitivo y posee un conocimiento informal de los términos creencias, deseos e intenciones; y su principal dificultad es saber cómo implementar eficientemente estas funciones (Rizo, Llorens, & Pujol, 2003).

1.9.2 Reactivas

Se basan en una estrecha relación entre percepción y acción, estos agentes no tienen un modelo simbólico del entorno, omitiendo la capacidad de realizar complejos procesos de razonamiento. Estas restricciones se deben a la creación de agentes compactos, tolerantes a fallos, y flexibles. A diferencia de otros tipos de agentes los agentes reactivos no obtienen su inteligencia de modelos internos, sino de la interacción con su entorno, por tal razón el proceso de interacción tiene gran importancia. Un agente reactivo no necesariamente debe tener una estructura compleja para actuar en un entorno complejo, basta con que observe el entorno y reconozca una serie de principios simples o dependencias. Este conocimiento es usado para desarrollar módulos específicos capaces de comprobar continuamente en su entorno la ocurrencia de ciertas situaciones específicas e iniciar una reacción directa cuando dichas situaciones ocurran. (Rizo, Llorens, & Pujol, 2003).

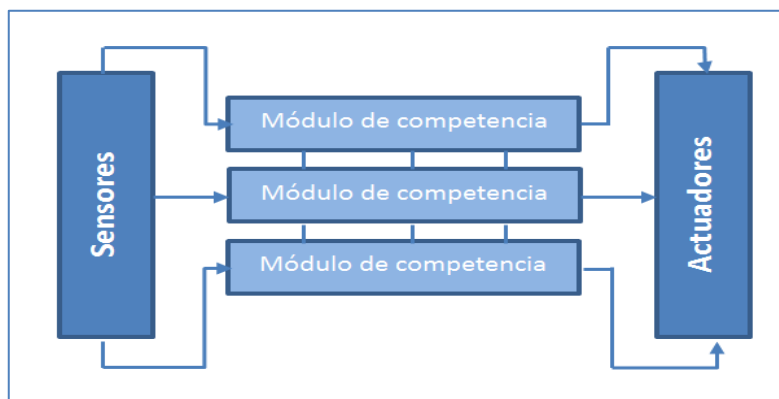


FIGURA 1.4 Arquitectura de Agentes Reactivos
Fuente:(Rizo, Llorens, & Pujol, 2003).

En la Figura 4, (Rizo, Llorens, & Pujol, 2003) nos presentan la arquitectura básica de los agentes reactivos que se corresponden con un sistema simple de estímulo/respuesta, los sensores se encargan de recoger la información de su entorno y reconocer la ocurrencia de situaciones cambiantes, para luego enviarla a los módulos de competencia correspondientes en formato de texto plano (no se usa representaciones simbólicas o lenguajes de comunicación de alto nivel), cada módulo de competencia tiene una tarea claramente definida pero no específicamente compleja, produciendo una reacción como salida en los mismos, que se transmite al exterior por medio de actuadores. La forma específica de los sensores y actuadores depende en gran medida de los objetos monitorizados por el sensor. Cabe recalcar que no existen componentes centrales como un planificador o un razonador, cada uno de los módulos debe tener todas las capacidades requeridas para realizar su tarea, por esta razón los agentes reactivos no pueden resolver tareas para las que no tengan un módulo de competencia. Los módulos trabajan en paralelo y su comunicación es directa (relación uno a uno entre dos componentes) entre ellos o a través del entorno, basada en mecanismos de comunicación simples, permitiendo así una rápida capacidad de reacción. Otro método de comunicación se produce cuando un módulo origina un cambio en su entorno y es observado por otro módulo, el cual inicia su proceso de respuesta, por esta razón este método es más lento, pero permite reaccionar a situaciones más complejas. Al trabajar los módulos de competencia en paralelo, varios de estos módulos pueden dispararse simultáneamente. Debido a su estructura descentralizada los módulos de competencia aumentan la tolerancia a fallos (si un módulo falla o no funciona correctamente los demás siguen trabajando) y la robustez. Los agentes reactivos habitualmente no tienen la capacidad de desarrollar planes, por lo cual sus tareas no son completamente determinadas; los planes pueden ser utilizados para optimizar el comportamiento de los agentes, pero en aplicaciones prácticas los agentes poseen información incompleta de su entorno y sus objetivos, ya que el entorno es muy dinámico y está sujeto a continuos cambios y por ende los objetivos son cambiante y los recursos necesarios para la creación de planes óptimos no suelen estar disponibles. Los módulos de competencia tienen objetivos implícitos definidos por su funcionalidad que no pueden ser cambiados, no pueden usar conocimiento interno para generar y seguir dinámicamente nuevos objetivos, a pesar de estas restricciones algunas personas opinan que los agentes reactivos son capaces de acciones orientadas al objetivo, similar a la inteligencia, ya que la orientación a objetivos resulta implícitamente de la interacción con el entorno y no de la planificación o evaluación centralizada.

1.9.3 Híbridas

Tanto las arquitecturas reactivas como las deliberativas presentan ciertas limitaciones, por esta razón se proponen los sistemas híbridos, que pretenden combinar aspectos de ambos modelos. La primera propuesta es construir un agente compuesto de dos subsistemas: uno deliberativo, que use un modelo simbólico y genere planes; y otro reactivo centrado en reaccionar a los eventos que tengan lugar en el entorno y no requieran mecanismos de razonamiento complejo. Debido a su naturaleza estas arquitecturas favorecen una estructuración por capas, que pueden ser (Corchado, 2005):

- ✓ **Vertical:** Una sola capa con acceso a sensores y actuadores.
- ✓ **Horizontal:** Todas las capas tienen acceso a sensores y actuadores.

Las capas se organizan jerárquicamente con información sobre el entorno a diferentes niveles de abstracción. La mayoría de las arquitecturas encuentran suficientes tres niveles (Corchado, 2005):

- ✓ **Reactivo** (nivel bajo), aquí se toman decisiones acerca de qué hacer en base a los estímulos recibidos en entorno en tiempo real.
- ✓ **Conocimiento** (nivel intermedio), se centra en el conocimiento que el agente tiene del medio, con ayuda de una representación simbólica del mismo.
- ✓ **Social** (nivel alto), se manejan los aspectos sociales del entorno, incluye tanto información de otros agentes como deseos, intenciones, etc.

El comportamiento global de los agentes está definido por la interacción entre estos niveles, la interacción cambia de una arquitectura a otra.

1.9.4 Comparación de arquitecturas

En la Tabla 1.3 se presenta las principales características de las Arquitecturas para construir agentes.

Tabla 1.3 Características Principales de las Arquitecturas

Arquitecturas	Características Principales
Agentes Deliberativos	<ul style="list-style-type: none"> ○ Elementos centralizados (planificador) ○ Representación interna del mundo exterior, esquema <i>estimulo -> manipulación simbólica (razonamiento lógico) -> respuesta</i> ○ No implica agente simple + entorno solo: Se pueden construir arquitecturas deliberativas para coordinar comportamiento. ○ Agentes complejos
Agentes Reactivos	<ul style="list-style-type: none"> ○ Sin elementos centralizados ○ Sin representación interna del entorno, esquema <i>estímulo -> respuesta</i>, con patrones de respuesta predefinidos. ○ No mantienen historia pasada ni estado ○ Agentes muy simples ○ Interacción con otros agentes muy sencilla
Agentes Híbridos	<ul style="list-style-type: none"> ○ Arquitectura en capas ○ Técnicas reactivas para bajo nivel y deliberativas para generar comportamientos complejos

Fuente:(Hernandez, 2003).

En la Tabla 3 se presenta las principales características de las Arquitecturas Deliberativa, Reactiva e Híbrida. En la Tabla 4 se muestra algunas ventajas e inconvenientes que presentan las diferentes arquitecturas.

Tabla 1.4 Comparativa entre las Principales características de las Arquitecturas

Arquitecturas	Comparativa
Agentes Deliberativos	<p>Ventajas:</p> <ul style="list-style-type: none"> ○ Tecnología familiar ○ Metodología clara ○ Muchas teorías adecuadas ○ Algunas facilitan la construcción de sistemas sociales y cooperativos en

	<p>los que existen muchos agentes autónomos.</p> <ul style="list-style-type: none"> ○ Se puede introducir fácilmente un mecanismo de aprendizaje. ○ Permiten la construcción de agentes realmente autónomos <p>Inconvenientes:</p> <ul style="list-style-type: none"> ○ Problemas para actualizar a tiempo el conocimiento, sobre todo en entornos de tiempo real altamente dinámicos en los que los recursos no están disponibles o son limitados. (aunque se han propuesto algunas variantes en las que se mejora este problema) ○ Problemas a la hora de traducir el entorno a una representación simbólica.
<p>Agentes Reactivos</p>	<p>Ventajas:</p> <ul style="list-style-type: none"> ○ Buen funcionamiento en entornos altamente dinámicos ○ Simplicidad ○ Robustez y tolerancia a fallos ○ Eficiencia, reacciones ante eventos muy rápidas. ○ Autonomía limitada <p>Inconvenientes:</p> <ul style="list-style-type: none"> ○ No hay consenso sobre la tecnología ○ No hay metodología y sólo algunas teorías aisladas ○ Es necesaria una gran cantidad de información local ○ El aprendizaje es problemático ○ Construcción por ensayo-error. Largo proceso de experimentación. ○ Los agentes reactivos sólo pueden ser utilizados para su propósito original ○ Interacciones entre agentes muy simples ○ La construcción de sistemas grandes es extremadamente complicada.
<p>Agentes Híbridos</p>	<p>Ventajas:</p> <ul style="list-style-type: none"> ○ Combinan las ventajas de arquitecturas deliberativas y las ventajas de arquitecturas reactivas.

Inconvenientes:

- Solución pragmática, pero ad-hoc
- No hay consenso claro pero si muchas similitudes
- No hay ni metodología ni teoría formal que la soporte
- Muy difíciles de generalizar
- Muy difícil reproducir sus resultados en diferentes dominios

Fuente:(Hernandez, 2003).

1.10 Arquitecturas multiagente

Según(Ferber, 1999), los sistemas multiagentes surgen ante la necesidad de desarrollar aplicaciones complejas, compuestas por varios subsistemas que interactúan entre sí, permitiendo distribuir la inteligencia entre varios agentes; lo cual permite la gestión inteligente de un sistema complejo coordinando distintos subsistemas que lo componen e integrándolo con los objetivos particulares de cada subsistema en un objetivo común. Dichos sistemas se emplean en varios casos, por ejemplo: cuando los problemas son físicamente distribuidos, cuando la complejidad de la solución requiere de experiencia muy heterogénea o cuando el problema a resolver está definido sobre una red de ordenadores. Actualmente la complejidad de la mayor parte de los problemas requiere una solución distribuida capaz de adaptarse a cambios en la estructura y en el entorno, así como la metodología de desarrollo que permita la construcción de todo un sistema a partir de distintas unidades autónomas.

Un sistema multiagente cooperante (Wesson & Hayes-Roth, 1980), presenta las siguientes características:

- Debe estar formado por un conjunto de agentes, manteniendo cada uno sus propias habilidades (adquisición de datos, comunicación, planificación y actuación).
- Un sistema multiagente tiene una misión común. Dicha misión puede dividirse en diferentes tareas independientes, de tal manera que puedan ejecutarse en paralelo. El sistema multiagente debe ser capaz de asignar a cada uno de sus componentes una o varias tareas concretas, tomando en cuenta cual es el objetivo común.
- Cada agente que compone el sistema tiene un conocimiento limitado (conocimiento de entorno, misión del grupo, intenciones de los otros agentes al realizar sus tareas).

- Los agentes del sistema poseen cierto tipo de especialización para realizar determinadas tareas, en función de: su conocimiento, capacidad de proceso y habilidad requerida.
- La distribución de las decisiones entre los distintos nodos del sistema permite:
 - Mantener la autonomía de cada agente (los agentes deciden según su propio entorno que tareas realizar).
 - Eliminar la necesidad de que toda la información del sistema se encuentre en un solo agente.
 - Descentralizar la toma de decisiones, permitiendo así que los sistemas sean más robustos ante posibles fallos.
 - Realizar acciones coordinadas, para esto es necesario un mecanismo de comunicación entre los agentes.
 - Crear una arquitectura de agentes dinámica, para que pueda adaptarse a cambios producidos en el entorno en un tiempo real.

Corchado (Corchado, 2005), en el problema de la coordinación entre los agentes, es igual de importante el proceso de razonamiento interno (toma de decisiones e identificación de información que se debe compartir) como el proceso de comunicación (cómo y cuándo debe producirse la comunicación entre nodos). La comunicación juega un papel muy importante al momento de definir un sistema multiagente.

(Ferber, 1999), distingue tres métodos de análisis en una organización compuesta por agentes:

- ✓ *Análisis funcional*, describe las diversas funciones del sistema multiagente, en sus diferentes dimensiones.
- ✓ *Análisis estructural*, distingue entre las diversas formas de organización e identifica los parámetros estructurales fundamentales.
- ✓ *Análisis de parámetros concretos*, centrado en las diferentes cuestiones que aparecen al pasar de una estructura abstracta a una organización concreta de agentes.

En el análisis estructural principalmente se necesita conocer qué relación existe entre los distintos agentes y las tareas que debe realizar el sistema (organización), es decir la relación que existe entre la función productiva del agente y la función de la organización.

Previo a este análisis debemos considerar al sistema multiagente desde el punto de vista de los agentes y las tareas que se deben realizar.

También se debe identificar la relación entre un agente y el problema, dicha relación puede verse desde: una perspectiva funcional (qué funciones realiza el agente y qué funciones debe realizar el sistema), la perspectiva de la distribución (los agentes están distribuidos en el espacio y se encargan cada uno de ellos de cierta parte del problema), y finalmente, una perspectiva orientada a objetos (considerando cada agente como un sistema que recibe mensajes y entrega productos ya terminados).

Para determinar las funciones que realiza el agente y las diversas tareas que debe realizar (distribución de tareas entre agentes y cuál redundancia existe) existen dos perspectivas: vertical, que analiza las funciones y su sucesiva descomposición y cómo se distribuyen estas tareas; y horizontal, que describe las funciones que son compartidas por los agentes. Estas perspectivas son complementarias y cada una se centra en una forma diferente de describir las tareas: la vertical analiza la descomposición de tareas y la horizontal describe las características de las tareas con parámetros comunes a todas las funciones que deben realizarse.

Cuando tenemos la descripción del sistema multiagente, es posible analizar las relaciones abstractas (describe las formas de interacción entre las diferentes clases de agentes) entre los agentes que existen en la organización, los enlaces de los agentes y las subordinaciones que existen entre los agentes para la toma de decisiones. Las relaciones abstractas son las siguientes(Ferber, 1999):

- *Relación de conocidos*, indica que dados dos agentes, uno tiene una representación del otro y conoce su dirección. Esta es la relación mínima entre dos agentes y es el soporte de otras relaciones.
- *Relación de comunicación*, un agente puede mandar mensajes a otro. El soporte para el canal de comunicación es la *relación de conocidos*.
- *Relación de subordinación*, describe la transferencia de una ejecución (tarea) entre dos agentes.
- *Relación operativa*, representa las dependencias entre tareas relacionadas.
- *Relación de información*, establece la validez de las dependencias entre las cosas que un agente conoce.
- *Relación de conflicto*, indica que los agentes tienen un conflicto por el acceso a los recursos y que es necesario coordinar las tareas mediante una negociación.

- *Relación competitiva*, corresponde con una competición entre los agentes y es una señal de que sus objetivos son incompatibles.

Las relaciones abstractas también pueden ser estáticas (establecida al definir la organización y no cambian con la ejecución) o dinámicas (cambian en función de la ejecución). La caracterización de la estática o dinámica tiene sentido únicamente en tres casos (Ferber, 1999):

- *Relación de subordinación estática o dinámica*, en la primera se define qué agente está subordinado a otro (relación maestro-esclavo). En la segunda existe una demanda de servicios por parte de un agente (petición). La petición puede o no ser aceptada por otro agente, lo que implica que la relación depende de la situación del sistema en cada momento.
- *Relación operativa estática o dinámica*, la estática define las dependencias de las tareas desde un inicio. La dinámica distribuye las tareas entre los agentes en función a su disponibilidad, básicamente la ejecución de una tarea dependiente está íntimamente ligada a que otro agente se haya comprometido a la ejecución de la tarea de la que depende.
- *Relación de información estática o dinámica*, en la estática existe una relación de confianza mutua definida estructuralmente. En la dinámica el agente tiene que demostrar la validez de la información, en consecuencia, la relación entre los agentes puede modificarse en función de cómo se modifique la confianza en el agente.

Los modos de enlace especifican cómo están unidos entre sí los agentes y su propia capacidad para automodificarse (cambiar sus roles). Los enlaces pueden ser fijos, variables y evolutivos. Los fijos no permiten la reorganización y adaptación al entorno de la organización y del agente convirtiendo al sistema en un programa (no útil al sistema multiagente). Los variables permiten una adaptación controlada de los agentes en algunos de sus parámetros, esta adaptación es la más usada en sistemas multiagente. Los evolutivos permiten que las relaciones y los propios agentes evolucionen sin definir los límites de la misma.

La relación de subordinación está definida por las estructuras jerárquicas e igualitarias. En las estructuras jerárquicas ciertos agentes tienen la capacidad de ordenar a los demás

agentes, de manera que controlan y coordinan todas las actividades. Mientras que en las estructuras igualitarias todos los agentes participan en la toma de la decisión final (si el enlace es fijo se utilizan como sistemas de distribución de tareas). Cuando diseñamos los agentes se puede definir la estructura de la organización a priori o a posteriori cuando surge durante la propia ejecución del sistema.

Las habilidades de cada uno de los agentes dependen del tipo de problema que se intente resolver. Los casos extremos son: que todos los agentes tengan todas las habilidades posibles o que cada agente tenga una única habilidad (especialista). Para poder analizar esta distribución se pueden usar dos índices: el grado de especialización y el grado de redundancia.

El grado de especialización evalúa la relación existente entre el número de habilidades del agente para un problema dado y el número de habilidades existentes para ese problema. El grado de redundancia evalúa la relación existente entre el número de agentes que tienen la misma habilidad y el número de agentes. A partir de estos parámetros se identifican cuatro puntos extremos (Ferber, 1999):

- *Organización no redundante e hiperespecializada*, cada agente tienen una habilidad que ningún otro agente la posee (descomposición funcional, cada función es un agente).
- *Organización redundante y especializada*, cada agente posee una sola habilidad y todos los agentes la poseen. Sirve para optimizar los problemas de ejecución cuando se usa varias máquinas para resolverlos.
- *Organización redundante y general*, cada agente tiene varias habilidades y varios agentes la poseen (sistemas multiagentes).
- *Organización no redundante y general*, cada agente tiene todas las habilidades y nadie más la posee (el sistema multiagente pasa a convertirse en un agente).

1.11 Plataforma de desarrollo del Sistemas Multiagente

1.11.1 JADE

Es un middleware¹ que facilita el desarrollo de aplicaciones multiagente (Peer-to-Peer), planificación de tareas del agente, herramientas gráficas para la monitorización logs, depuración. Esta desarrollado en JAVA y distribución Open Source, además cumple con las especificaciones FIPA (Bellifemine, Caire, & Greenwood, 2007).

¹Middleware, software asistente de una aplicación para interactuar con otras aplicaciones.

El estándar PIFA permite la interoperabilidad entre plataformas de diferentes empresas y organizaciones, especifica la normativa que se llevar a cabo para la construcción de plataformas Multiagentes y no restringen la tecnología usada para la implementación de plataformas.

Características

- ✓ Plataforma distribuida.
- ✓ Herramientas de debugging.
- ✓ Movilidad de agentes inter-plataforma.
- ✓ Soporta ejecución paralela de múltiples agentes.
- ✓ Transporte de mensajes ACL (Lenguaje de Comunicación de Agentes) dentro de la plataforma.
- ✓ Paso asíncrono de mensajes.
- ✓ Interface para aplicaciones externas.
- ✓ Arquitectura peer to peer.
- ✓ Portabilidad, desarrollado en JAVA.
- ✓ Seguridad distribuida, tolerancia a fallos, soporte para la réplica de agentes y servicios, persistencia.
- ✓ JADE y .NET
- ✓ JADE, Protege, XML, RDF y OWL

Una aplicación basada en JADE consta de componentes activas denominadas “Agentes”, cada agente tienen un nombre identificativo único y es un participante (peer) capaz de comunicarse bidireccionalmente con otros agentes.

1.12 Técnicas para la extracción de emociones

1.12.1 Extracción de emociones a través de texto

Las técnicas empleadas para el análisis de documentos tienen como objetivo la detección y clasificación de emociones presentes en los textos. Entre las técnicas empleadas tenemos las siguientes:

- Basada en recuperación de información
- Basada en clasificación de textos
- Basada en diccionario afectivo
- Técnicas estadísticas para el reconocimiento de emociones:
 - Naïve Bayes

- Basada en PLN (Procesamiento de Lenguaje Natural)
 - Segmentación
 - Lematización

✓ **Basada en recuperación de información**

Primero debemos realizar un análisis PMI-IR (Pointwise-Mutual Information and Information Retrieval, PMI permite calcular la fuerza de la asociación semántica entre palabras y luego, se aplica recuperación de información (IR) para obtener las estadísticas de la co-ocurrencia de palabras).

Turney en el año 2002, en su investigación describe un clasificador no supervisado basado en la opinión que decide el carácter positivo o negativo de un documento en base a la orientación semántica de los términos que aparecen en el mismo. Esta orientación se calcula mediante el algoritmo PMI-IR anteriormente mencionado, que consiste en estimar la información Mutua Puntual (Pointwise Mutual Information) entre el término en cuestión y un par de palabras “semilla” que sirven de representantes inequívocos. La idea de este cálculo de orientación semántica es que expresiones que indiquen una opinión positiva aparecerán con mayor frecuencia cerca de una palabra con claras connotaciones positivas como excelente y con mucha menor frecuencia cerca de una palabra con connotaciones negativas como horrible, es decir, palabras que expresen un sentimiento parecido a menudo co- aparecen en un mismo texto, mientras que palabras que expresan un sentimiento contrapuesto raramente aparecen juntas. Así, una palabra que se encuentre de forma más frecuente junto a excelente que junto a horrible se estimara como una palabra de orientación positiva(Martín, 2012).

✓ **Basada en clasificación de texto**

La técnica más utilizada dentro del ámbito de clasificación temática es la basada en Máquinas de Vectores de Soporte (SVM), esta técnica también es empleada en el desarrollo de muchos clasificadores emocionales como es el caso de (Leshed & Kaye, 2006), donde se presenta un clasificador emocional de blogs que utiliza SVM.

En (Turney & Littman, 2003) se presenta un sistema de identificación de la polaridad del texto basado en Latent Semantic Analysis (LSA). Para saber la polaridad de cada palabra

del texto, se calcula la diferencia entre su similitud con un conjunto de palabras positivas y otro de palabras negativas (Martín, 2012).

✓ **Basada en diccionario afectivo**

Esta técnica se basa en buscar las palabras afectivas que contenga el texto en un diccionario de vocablos afectivos construido previamente. Enfatiza Emotional Keyword Spotting (EKS), debido a su sencillez de implementación. La emoción global del texto se determina a partir de la media de los valores emocionales de cada una de las palabras clave detectadas.

Una extensión de EKS es la denominada afinidad léxica, que exporta la emoción de las palabras clave a sus palabras cercanas (Liu, Lieberman, & Selker, 2003). Ambas son incapaces de detectar cambios de polaridad de la emoción debido a elementos del texto.

✓ **Técnicas estadísticas para el reconocimiento de emociones**

- **Naïve Bayes**

Es considerado como parte de los clasificadores probabilísticos, basado en la suposición que las cantidades de interés se rigen por distribuciones de probabilidad y que la decisión óptima puede tomarse por medio de razonar acerca de esas probabilidades en conjunto con los datos observados (Mitchell, 1997). A continuación se describe el Naïve tradicional:

El clasificador es construido usando Tr para estimar la probabilidad de cada clase, cuando una nueva instancia es presentada i_j , el clasificador le asigna la categoría $c \in C$ más probable por aplicar la regla (Téllez, 2005):

$$c = \arg \max_{c_i \in C} P(c_i | i_j) \quad (1)$$

Utilizamos el teorema de Bayes para estimar la probabilidad y se obtiene:

$$c = \arg \max_{c_i \in C} \frac{P(i_j | c_i) P(c_i)}{P(i_j)} \quad (2)$$

El denominador puede ser omitido:

$$c = \arg \max_{c_i \in C} P(i_j | c_i) P(c_i) \quad (3)$$

Considerando que el esquema es llamado “Naïve” por su supuesta independencia entre atributos, se asume que las características son condicionalmente independientes dadas las clases. Esto produce la simplificación de los cálculos y tenemos:

$$c = \arg \max_{c_i \in C} P(c_i) \prod_{k=1}^n P(a_{kj} | c_i) \quad (4)$$

Donde $P(c_i)$ es la fracción de ejemplos en Tr y pertenece a la clase c_i , y $P(a_{kj} | c_i)$ se calcula de acuerdo al teorema de Bayes.

Naïve Bayes consiste en construir una hipótesis por medio de estimar las diferentes probabilidades $P(c_i)$ y $P(a_{kj} | c_i)$ en términos de sus frecuencias sobre Tr .

✓ **Basada en PLN (Procesamiento de Lenguaje Natural)**

- **Segmentación**

La segmentación de texto consiste en dividir un texto en unidades indivisibles denominadas tokens. La segmentación es lo primero que se utiliza para el procesamiento de texto. La lista generada a partir de los tokens se utiliza como entrada de un vocabulario.

- **Lematización (Stemming)**

La Lematización es un proceso lingüístico que consiste en la transformación de una palabra flexionada² a su lema correspondiente. El lema de una palabra es el representante de las palabras flexionadas, es decir, es la palabra que encontramos como entrada en un diccionario o una parte de la palabra luego de eliminar sus afijos y de la cual se desprende todas las formas flexionadas de la misma palabra.

² Una palabra conjugada, con género, plural/singular, etc.

Consiste en extraer la auténtica raíz de una palabra a partir de un diccionario. Cabe recalcar que la Lematización no depende solamente de la forma flexionada, sino también de su contexto.

1.12.2 Extracción de emociones a través de Texto e Imagen

Las principales técnicas empleadas para el reconocimiento automático de emociones en texto e imágenes son las siguientes:

✓ Técnicas estadísticas para el reconocimiento de emociones:

- Métodos Supervisados de Clasificación
- Máquina de Vector Soporte
- Árboles de Clasificación
- Redes Bayesianas
- Algoritmo de votación: Bagging, Boosting
- Modelos Markovianos Ocultos
- Métodos no Supervisados
- Redes Neuronales

✓ **Técnicas estadísticas para el reconocimiento de emociones**

- **Métodos Supervisados de Clasificación**

Según (Cabello, Conde, Martín de Diego, & Serrano, 2006), la clasificación es el proceso de asignar objetos a un conjunto prefijado de categorías o clases, en nuestro caso las categorías están determinadas por las emociones que queremos detectar a partir de una serie de características medidas sobre los individuos. Cuando se a priori el conjunto de clases (emociones), este proceso de clasificación se lo conoce con el nombre de clasificación supervisada, mientras que, si el conjunto de clases es desconocido el proceso recibe el nombre de clasificación no supervisada.

Si existe una función desconocida que va del espacio de características al espacio de emociones, nos referimos a ella como función objetivo. La solución al problema de clasificación es la estimación de la función objetivo (un clasificador). Permitiendo describirlo de la siguiente manera: dada una muestra de objetos (caras, textos, discurso, etc...), encontrar una función que asigne cada objeto a una de las emociones predefinidas, de manera que se minimice el error promedio de clasificación para futuras observaciones.

Este algoritmo de clasificación es también conocido como proceso de aprendizaje, reconocimiento de patrones o discriminación. Hay dos etapas primordiales en el diseño de un clasificador: la fase de entrenamiento y la fase de validación.

- **Máquinas de Vector Soporte (SVM)**

El fundamento principal del algoritmo SVM, es la capacidad de separar en dos clases un conjunto de datos (Vapnik, 1999). Puede ser aplicado a problemas de clasificación y regresión en una infinidad de dominios, debido a su versatilidad posee diversas opciones de configuración (Martínez, Martín, Perea, & Ureña, 2011).

Dado un conjunto de datos que pertenecen a dos clases, la idea básica de SVM es encontrar un hiperplano que puede distinguir dos clases diferentes, el hiperplano es decidido por el margen máximo de dos clases y las muestras que se encuentran en el margen se denominan “vectores soporte”. La ecuación del hiperplano se describe como (Chuang & Wu, 2004):

$$D(x) = \sum_{i=1}^N \alpha_i y_i (x \cdot x_i) + w_0$$

Tradicionalmente SVM puede construir un límite de difícil decisión sin probabilidad de salida. Por esta razón se propone un SVM con una probabilidad continua de salida dada una muestra de prueba x' , la probabilidad de que x' pertenece a la clase C es $P(\text{clase } c|x')$. Este valor es calculado en base a los siguientes factores:

- La distancia entre la prueba de entrada y el hiperplano, reconocimiento de las emociones multimodal de voz y texto.

$$R = \frac{D(x') \|w\|}{1/\|w\|} = D(x')$$

- La distancia de la clase central al hiperplano.

$$R' = \frac{R}{D(\bar{x})} = \frac{D(x')}{D(\bar{x})}$$

Donde \bar{x} es el centro de los datos de ensayo de una clase.

- La confianza de la clasificación de la clase, la precisión de la clasificación evaluada en base a los datos de entrenamiento se usa para definir la confianza de la clasificación de la clase C.

$$P_c = \frac{\text{Número de sentencias correctamente reconocidas como clase } c}{\text{Número total de sentencias en la clase } c}$$

La probabilidad de salida está definida de la siguiente manera de acuerdo a los factores anteriores:

$$P(\text{clase}_c|x') = \frac{P_c}{1 + \exp(1 - R')} = \frac{P_c}{1 + \exp\left(1 - \frac{D(x')}{D(\bar{x})}\right)}$$

El resultado final es la combinación de diferentes salidas:

$$\begin{aligned} P(\text{clase}_c|x') &= \left(\prod_{i=1}^s P_i(\text{clase}_c|x') \right)^{\frac{1}{s}} \\ &= \left(\prod_{i=1}^s \left(\frac{P_c}{1 + \exp(1 - D(x')/D(\bar{x}))} \right) \right)^{\frac{1}{s}} \end{aligned}$$

Donde la probabilidad $P_i(\text{clase}_c|x')$ es la salida de SVM.

- **Árboles de clasificación**

Los árboles de clasificación (decisión o de identificación), es uno de los métodos de aprendizaje inductivo supervisado no paramétrico más utilizado, destacado por su sencillez, pese a carecer de expresividad de las redes semánticas o de la lógica de primer orden, su dominio de aplicación no está restringido a un ámbito específico, sino que puede usarse en diversas áreas: juegos, diagnóstico médico, predicción meteorológica, control de calidad, etc. (Cortijo, 2000).

Un árbol de clasificación es una forma de representar el conocimiento obtenido en el proceso del aprendizaje inductivo. Puede verse como la estructura resultante de la partición recursiva del espacio de representación a partir del conjunto de prototipos. Esta partición

recursiva se traduce en una organización jerárquica del espacio de representación que puede modelarse mediante una estructura de tipo árbol. Cada nodo interior contiene una pregunta sobre un atributo concreto (con un hijo por cada posible respuesta) y cada nodo hoja se refiere a una decisión (clasificación).

La clasificación de patrones se realiza en base a una serie de preguntas sobre los valores de sus atributos, empezando por el nodo raíz y siguiendo el camino determinado por las respuestas a las preguntas de los nodos internos, hasta llegar a un nodo hoja. La etiqueta asignada a esta hoja es la que se asignaría al patrón a clasificar.

La metodología a seguir se resume en dos pasos:

1. **Aprendizaje**, consiste en la construcción del árbol a partir de un conjunto de prototipos, S . Es la fase más compleja y determinante para el resultado final.

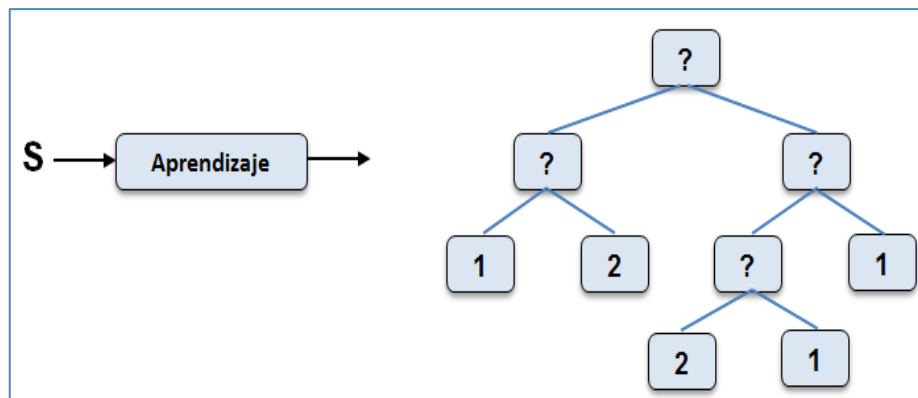


FIGURA 1.5 Aprendizaje con un árbol de decisión
Fuente: (Cortijo, 2000).

2. **Clasificación**, es el etiquetado de un patrón, X , independiente del conjunto de aprendizaje. Se trata de responder a las preguntas asociadas a los nodos interiores utilizando los valores de los atributos del patrón X . Este proceso se repite desde el nodo raíz hasta alcanzar una hoja, siguiendo el camino impuesto por el resultado de cada evaluación.

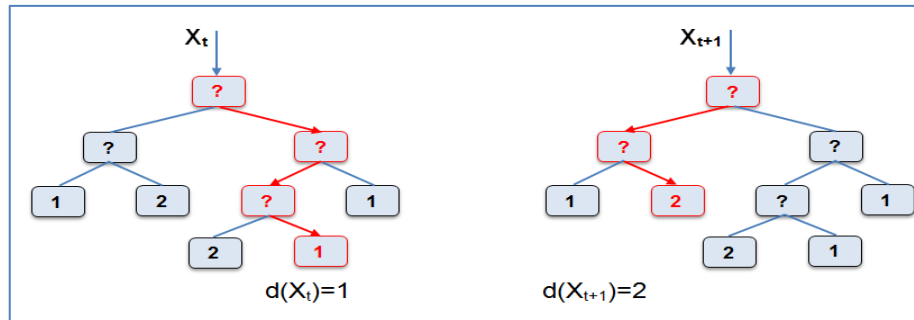


FIGURA 1.6 Clasificación con un árbol de decisión
Fuente:(Cortijo, 2000).

- **Redes Bayesianas**

Las redes bayesianas son clasificadores empleados para representar distribuciones conjuntas de modo que permitan calcular la probabilidad a posteriori de un conjunto de clases (emociones) dado un conjunto de características observadas en los objetos, y así clasificar los objetos en la clase más probable. Una red bayesiana está compuesta por un grafo dirigido en el cual cada nodo está asociado con una característica y con una distribución de probabilidad condicional. El grafo representa la estructura y las distribuciones de probabilidad los parámetros de la red. La idea general consiste en usar una estrategia que pueda buscar de modo eficiente en el espacio de posibles estructuras y extraer aquella que dé mejores resultados de clasificación (Cabello, Conde, Martín de Diego, & Serrano, 2006).

- **Algoritmo de Votación: Bagging, Boosting**

El método llamado Bagging (“bootstrap aggregating”) fue propuesto por Leo Breiman para clasificación y árboles de regresión (Breiman, 1994). Las predicciones Bagging es un método para generar múltiples versiones de un predictor y el uso de estos para obtener un predictor agregado, supongamos que disponemos de un modelo, acorde a nuestra muestra de objetos, obteniendo así un clasificador asociado a cada objeto. Repetimos la estimación del clasificador, modificando en cada caso la muestra de entrenamiento. Cada una de estas muestras recibe el nombre de “muestra de bootstrap”. El método Bagging consiste en obtener una media de las predicciones sobre un conjunto de muestras bootstrap (Freund & Schapire, 1996). El clasificador final se obtiene a partir de una ponderación de clasificadores “débiles”, es decir, con poca capacidad de generalización (Cabello, Conde, Martín de Diego, & Serrano, 2006).

- **Modelos Markovianos ocultos**

Los modelos Markovianos ocultos, asumen que el sistema estudiado sigue un proceso de Markov con parámetros desconocidos, la tarea principal consiste en determinar los parámetros ocultos a partir de los parámetros observados. La principal diferencia respecto a un modelo de Markov habitual consiste en que los estados no son directamente visibles para el observador, pero sí lo son las variables influenciadas por el estado. Cada estado tiene una distribución de probabilidad asociada sobre el conjunto de posibles valores de salida. La secuencia de valores de salida generados a partir de un HMM (Hidden Markov Model), nos dará cierta información sobre la secuencia de estados. También se debe considerar, que existen tres problemas fundamentales a resolver en el diseño de un modelo HMM y son: la evaluación de la probabilidad o verosimilitud de una secuencia de observaciones dado un modelo HMM específico; la determinación de la mejor secuencia de estados del modelo; y el ajuste de los parámetros del modelo que mejor se ajusten a los valores observados (Cabello, Conde, Martín de Diego, & Serrano, 2006).

- **Métodos no supervisados**

Es posible aplicar técnicas de clasificación en las cuales no se emplea el conocimiento de las emociones en la muestra de entrenamiento, este tipo de técnicas reciben el nombre de métodos no supervisados. En la clasificación no supervisada diremos que hemos obtenido una buena clasificación cuando los grupos creados sean homogéneos respecto a los individuos que los forman y heterogéneos entre sí. Según el ambiente del problema de reconocimiento de emociones, este tipo de técnicas serán útiles cuando las bases de datos disponibles se recojan en entornos no controlados y los individuos reflejen emociones no indicadas a priori por el investigador (Cabello, Conde, Martín de Diego, & Serrano, 2006).

- **Redes Neuronales**

Una red neuronal artificial es un sistema de procesamiento de información que tiene ciertas aptitudes en común con las redes neuronales biológicas. Las redes neuronales artificiales se han desarrollado como generalizaciones de los modelos matemáticos de cognición humana o neuronas biológicas, basado en los supuestos de que (Fausett, 1994):

- El procesamiento de la información ocurre en muchos elementos simples llamados neuronas.
- Las señales son transferidas entre neuronas a través de enlaces de conexión.
- Cada enlace de conexión tiene un peso asociado, el cual, en una red neuronal típica, multiplica la señal transmitida.
- Cada neurona aplica una función de activación (usualmente no lineal) a su entrada de red (suma de entradas pesadas) para determinar su señal de salida.

Una red neuronal consta de varias capas de unidades de procesamiento o neuronas interconectadas. La capa de entrada recibe términos, mientras que las unidades o neuronas de la capa de salida mapean clases o categorías. Cada neurona acepta como entrada las salidas procedentes de otras neuronas, siendo la entrada efectiva a la neurona la suma ponderada de las entradas reales a dicha neurona. Cada neurona se caracteriza por su estado de activación, que es un valor que oscila entre 0 (no activada) y 1. Cada neurona realiza una tarea sencilla: recibe la información de entrada de las neuronas o del exterior y la usa para calcular una señal de salida que se propaga a otras unidades.

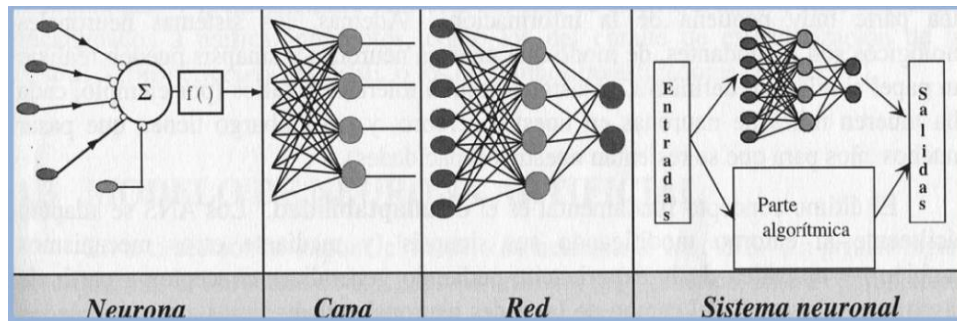


FIGURA 1.7 Descomposición de una red neuronal
Fuente:(Martín, 2012)

1.12.3 Modelado de la Frustración

En la publicación de (McQuiggan, Lee, & Lester, 2007) nos manifiesta que para crear modelos que nos ayuden a realizar predicciones exactas sobre la frustración en los estudiantes lo más pronto posible, primero debemos recopilar datos de ensayo mediante la observación de la interacción de los estudiantes con un sistema de tutorías inteligente.

A partir de estos datos de ensayo inducimos modelos n-gram para hacer predicciones tempranas de la frustración de los estudiantes. Los modelos basados en modelos n-gram son útiles para la predicción temprana debido a que son inducidos a partir de secuencias de

observación, haciendo predicciones con cada nueva observación hasta que llegamos a la observación final de la secuencia.

En la última observación se obtiene evidencia del estado emocional del estudiante. Cada predicción del modelo n-gramas intenta determinar el estado afectivo del estudiante registrado en la observación final de la secuencia. En muchos casos, las predicciones del modelo n-gram convergirán anticipadamente en el correcto estado afectivo en una secuencia de observaciones. El punto en el cual un modelo n-gram empieza a hacer la primera predicción correcta y luego continua haciendo predicciones correctas al resto de la secuencia se lo conoce como punto de convergencia.

Mientras que los modelos secuenciales, tales como n-gram nos permiten realizar predicciones iniciales, no son computacionalmente muy adecuados para el procesamiento de grandes datos multidimensionales.

Para solucionar este problema, investigaron tres técnicas de modelado no secuenciales: naïve Bayes, árboles de decisión, y máquinas de vectores de soporte. Para permitir a los modelos no-secuenciales hacer predicciones tempranas, utilizamos los resultados de los modelos de n-gram. Utilizando información del punto de convergencia, construimos un conjunto de datos que contienen solo aquellas observaciones que preceden al punto de convergencia. Así, los modelos inducidos a partir de datasets recién contruidos son capaces de hacer predicciones afectivas del estudiante mucho antes de la observación en la cual se registró el estado afectivo del estudiante.

- **Modelos n-gram para la predicción temprana de la frustración**

Considerando una secuencia de observaciones O_1, O_2, \dots, O_n , el objetivo del reconocimiento afectivo es identificar el estado afectivo más probable del estudiante E^* (frustrado o no frustrado) tal que:

$$\begin{aligned} E^* &= \arg \max P(E|O_1, O_2, O_3, \dots O_n) \\ &= \arg \max P(E|O_{1:n}) \end{aligned}$$

Donde cada O_i es una observación que codifica los objetivos del usuario, la acción del usuario, la ubicación donde se realizó la acción y las respuestas fisiológicas como la

frecuencia cardiaca y las respuestas galvánicas de la piel. La secuencia de observación O_1, O_2, \dots, O_n , se denota por $O_{1:n}$. Aplicando la regla de Bayes y la regla de la cadena, la ecuación se convierte en:

$$E^* = \arg \max P(O_n | O_{1:n-1}, E) P(O_{n-1} | O_{1:n-2}, E) \\ \cdot P(O_{n-2} | O_{1:n-3}, E) \dots P(O_1 | E) P(E)$$

Sin embargo, la estimación de estas probabilidades condicionales es poco práctica, requiere exponencialmente grandes conjuntos de datos de entrenamiento, por lo que hacer una suposición de Markov que una observación O_i depende únicamente del estado afectivo E y una ventana limitada de las observaciones precedentes.

Exploramos dos modelos n-gram de reconocimiento afectivo para detectar la frustración del estudiante, un modelo unigrama y un modelo bigrama. El modelo unigrama se basa en la suposición de que, dado el estado afectivo E , O_i es condicionalmente independiente de todas las demás observaciones. Por lo tanto, la fórmula de reconocimiento afectivo para el modelo unigrama puede ser simplificada a:

$$E^* = \arg \max P(E) \prod_{i=1}^n P(O_i | E)$$

El modelo de bigramas se basa en la suposición de que, dado un estado afectivo E y las observaciones precedentes O_{i-1} , O_i es condicionalmente independiente de todas las demás observaciones. Por lo tanto, la fórmula de reconocimiento afectivo para el modelo bigrama se puede simplificar a:

$$E^* = \arg \max P(E) \prod_{i=1}^n P(O_i | O_{i-1}, E)$$

La fórmula resultante para los modelos unigrama y bigrama son muy eficientes porque la actualización de la predicción afectiva para cada nueva observación solo requiere calcular el producto de la probabilidad devuelta por la predicción anterior y la probabilidad condicional actual.

1.13 Herramientas

1.13.1 Herramientas para la extracción de emociones en texto

Se ha considerado el estudio de varias herramientas, con la finalidad de determinar qué herramienta es la más adecuada para el análisis de texto y extracción de emociones, para esto se realizó un cuadro comparativo de las herramientas y de acuerdo a su funcionalidad se obtuvo las siguientes características descritas en la Tabla 1.5.

Tabla 1.5 Comparación de herramientas para extracción de emociones en Texto

Características Evaluadas	Alchemy API Sentiment Analysis	SentiStrength	Sentiment Analysis NLTK
Formatos de Entrada	Texto desde HTML Texto de una URL Texto simple	Texto simple palabras clave De un dominio específico	Archivos .txt, xls, pdf, MSWord, XML, formatos binarios
Formatos de Respuesta	XML, JSON, RDF	Txt	Txt, Fcfg
Idiomas soportados	English, German	Finlandés, alemán, neerlandés español, ruso, portugués, francés, árabe, polaco, persa, sueco, griego, galeses, italiano, turco.	10 idiomas entre ellos español, catalán, portugués y euskera.
Transacciones por día	<ul style="list-style-type: none"> • Para usuarios académicos aprobados se les permite hacer un máximo de 30.000 llamadas a la API de cada día. • 1.000 llamadas diarias para uso comercial • Con un Plan de Suscripción AlchemyAPI hasta 200 millones al día. 	Sin restricciones	Sin restricciones

Lenguaje (s) de desarrollo de SDK	Desarrollado en 7 lenguajes como: Android OS, Java, Perl, Rubí, Python (2.x), PHP-5, C/C++ y C# (.NET)	Java, Python, Ruby	Python
Usa PNL Procesamiento de Lenguaje Natural	Si	Si	Si
Key para usar el API	Si	No	No
Distribución gratuita	Si	No (£ 1000)	Si
Extrae metadatos semánticos	Si	No	Si
Evalúa EmotionLookUp Table	Si	Si	Si
Se adecua diccionarios	No	Si	Si
Gráficos que representan el sentimiento	No	No	Si
Procesamiento en tiempo real	Si	Si	Si

Para la evaluación de las herramientas contenidas en la **Tabla 1.5**, se tomó como ejemplo el siguiente texto, obtenido de la interacción del estudiante (mensaje) de un curso virtual:

Saludos cordiales Ing.: Acabo de visualizar mis notas en el Eva, y tengo la preocupante inquietud en su materia que tengo en trabajo el puntaje de 5,42; tengo entendido que esta nota es sobre 12 y el examen presencial sobre 28; por lo que me veo con una calificación super baja, y mi admiración es porque entregue todo, trabajos, tareas, foros, inclusive el trabajo final envíe antes de la fecha solicitada por cuanto sali fuera del país; justamente esta es la única materia que me falta conocer las notas, las 5 otras materias que tome ya las pase con buenos resultados; Ud. entenderá es por ello mi gran preocupación, cosa que no me ha sucedido con ninguna otra materia. En espera de su oportuna respuesta le anticipo mis agradecimientos.

- **AlchemyAPI**

AlchemyAPI utiliza la tecnología de Procesamiento de Lenguaje Natural (PLN) y algoritmos de aprendizaje automático para extraer metadatos con contenido semánticos. Proporciona mecanismos fáciles de usar para identificar sentimientos, clasificándolos en: Positivo y Negativo. AlchemyAPI clasifica automáticamente, etiquetas y enriquece cualquier contenido de texto y en la web. Ofrece soporte multilingüe y avanzadas características de desambiguación que no se encuentran en ninguna otra solución.

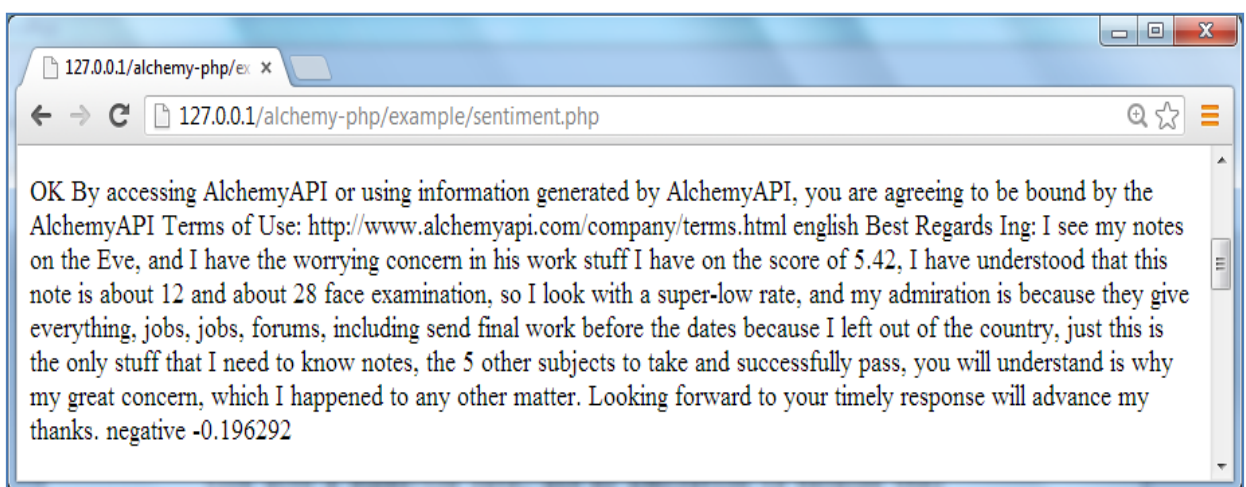


FIGURA 1.8 Resultados de Polaridad del Mensaje con AlchemyAPI

Para poder obtener la polaridad del mensaje fue necesario transcribirlo al idioma Inglés, y se obtuvo una polaridad negativa equivalente a -0,196292.

- **SentiStrength**

SentiStrength calcula la fuerza de los sentimientos positivos y negativos en los textos cortos, incluso para un lenguaje informal. Tiene una precisión de nivel humano de los textos, excepto textos políticos. Los resultados pueden ser positivo/negativo o positivo/negativo/neutral. SentiStrength mide los sentimientos en base a dos parámetros:

- -1 (No negativo) a -5 (muy negativa)
- 1 (no positivo) a 5 (muy positivo)

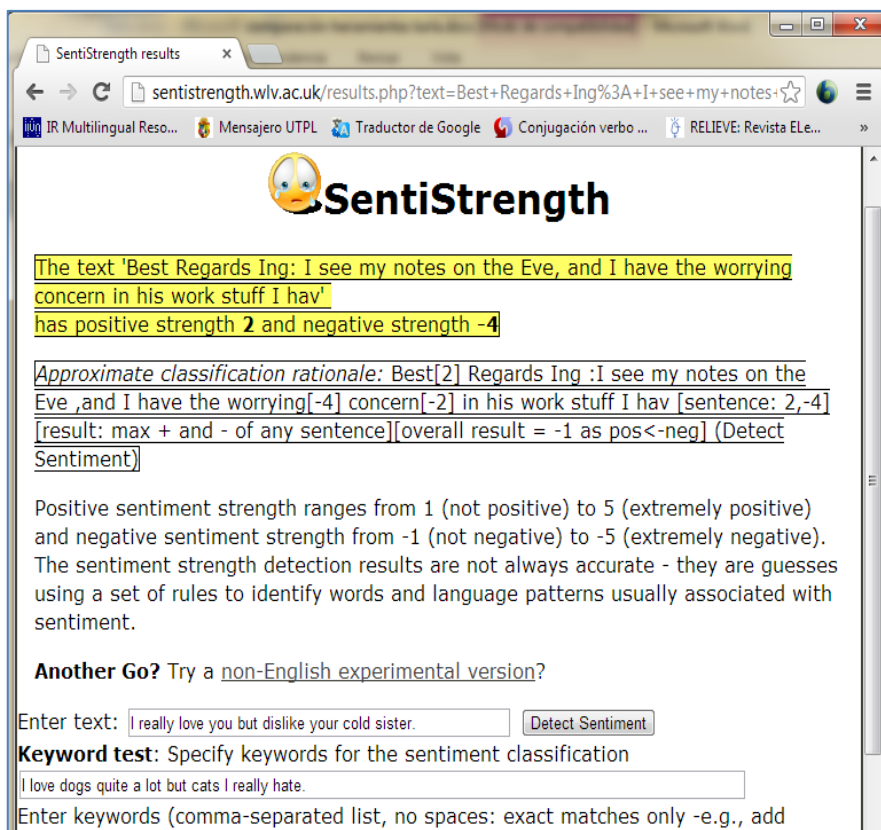


FIGURA 1.9 Resultados de Polaridad del Mensaje con SentiStrength

El resultado obtenido nos manifiesta que el contenido del mensaje tiene una polaridad de: 2 (+) y 4 (-). Obteniendo una polaridad Negativa.

- **Sentiment Analysis NLTK**

Proporciona una interfaz fácil de usar, con más de 50 corpus y los recursos léxicos, como WordNet, junto con un conjunto de bibliotecas de procesamiento de textos para la clasificación, simbolización, derivado, etiquetado, análisis sintáctico y razonamiento semántico. Cabe recalcar que NLTK es simple, extensible, modularizado, rápido y presenta gran cantidad de librerías para: expresiones regulares, gramáticas, parsers, modelos estadísticos, clasificadores. Además entre las herramientas que posee están: tokenizador de palabras, tokenizador de oraciones, etiquetador gramatical, chunkers, reconocedor de entidades, expresiones regulares, gramáticas, parsers, stemmers o lemmatizers, wordnet, algoritmos de aprendizaje automático, grandes colecciones de texto (Corpus).

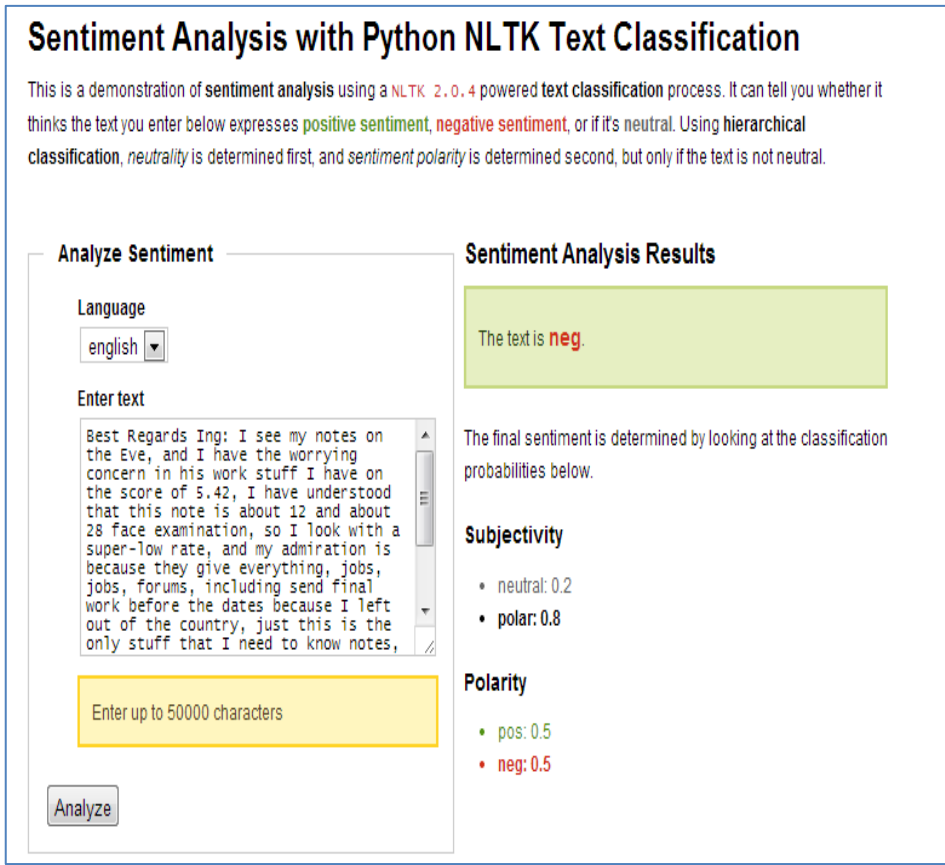


FIGURA 1.10 Resultados de Polaridad del Mensaje con NLTK

Como resultado del análisis se obtuvo los siguientes resultados:

Subjetividad: neutral de 0.2 y polar de 0.8

Polaridad: positiva de 0.5 y negativa de 0.5

Lo que nos permite determinar con mayor precisión que el texto tiene una polaridad **negativa**.

1.13.2 Herramientas para la extracción de emociones en video

Para la evaluación de las herramientas de análisis de video contenidas en la **Tabla 1.6**, se tomó como ejemplo una imagen (foto) y una imagen en tiempo real.

Tabla 1.6 Comparación de Herramientas para extracción de emociones en Video

Características Evaluadas	nVisio	EmotionRecognition
Formatos de Entrada	Permite una carpeta de imágenes a procesar (jpg, .jpeg, or .png.)	Captura visual desde WebCam
Formatos de Respuesta	Archivo CSV, Emotion ML (Content-Type: application / xml), JSON (Content-Type: application / json) o JSONP (Content-Type: application / javascript).	Html
Idiomas soportados	Ingles	Ingles
Transacciones por día	Sin restricciones	Sin restricciones
Lenguaje (s) de desarrollo de SDK	Python, PHP, Java / AndroidyiOS.	No dispone SDK, desarrollado en Java
Usa PNL Procesamiento de Lenguaje Natural aprendizaje de máquinas y otros métodos	Si	Si
Key para usar el API	Si	No
Distribución gratuita	No	No
Gráficos que representan el sentimiento	Si	Si
Procesamiento en tiempo real	Si	Si

- **Nvisio**

Nvisio proporciona un API, diseñado para procesar en tiempo real imágenes faciales, fotografías o fotogramas de vídeo desde webcam. Extrae los rasgos faciales básicos alrededor de los ojos, cejas, nariz, boca y barbilla y un perfil de la emoción, que consiste en siete medidas emocionales universales: felicidad, sorpresa, tristeza, asco, miedo, ira, y neutral.

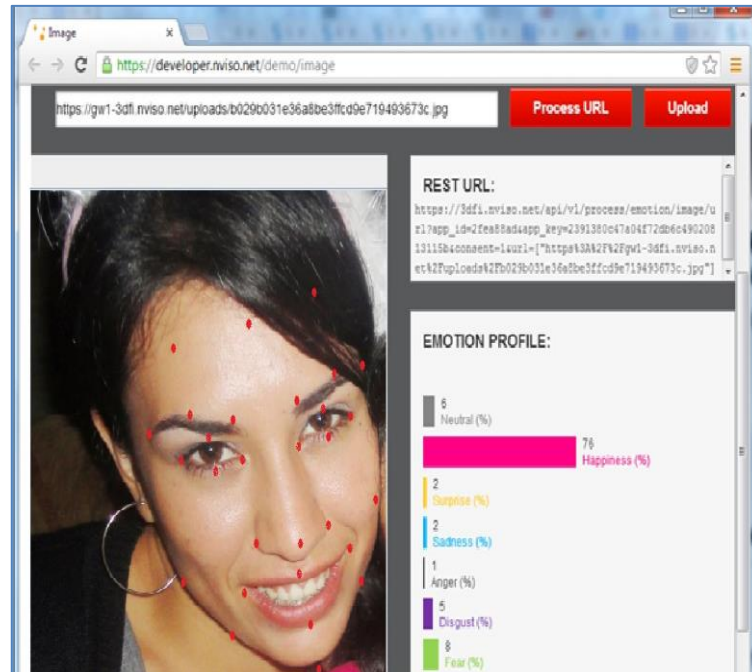


FIGURA 1.11 Resultados de Polaridad de la Imagen en Nvisio

Luego de procesar la imagen se obtuvo las siguientes emociones resultantes:

- ✓ Neutral 6%
- ✓ Happiness 76%
- ✓ Surpose 2%
- ✓ Sadness 2%
- ✓ Anger 1%
- ✓ Disgust 5%
- ✓ Fear 8%

Obteniendo como resultado del análisis de la imagen un 76% de felicidad, es decir, su polaridad positiva.

- **EmotionRecognition**

Analiza de forma automática en tiempo real las expresiones faciales de los seres humanos en videos digitales. Basado en el contenido visual digital, categoriza las emociones (Feliz, Neutro, Triste, Miedo, Enojo, Sorprendido, Disgustado). El sistema al funcionar en tiempo real se ejecuta en la PC y utiliza la cámara web, es sensible a los cambios de iluminación, movimientos de cabeza, la oclusión parcial de la cara (por ejemplo. Mover la mano temporalmente delante de la cara) (Gevers, 2008).

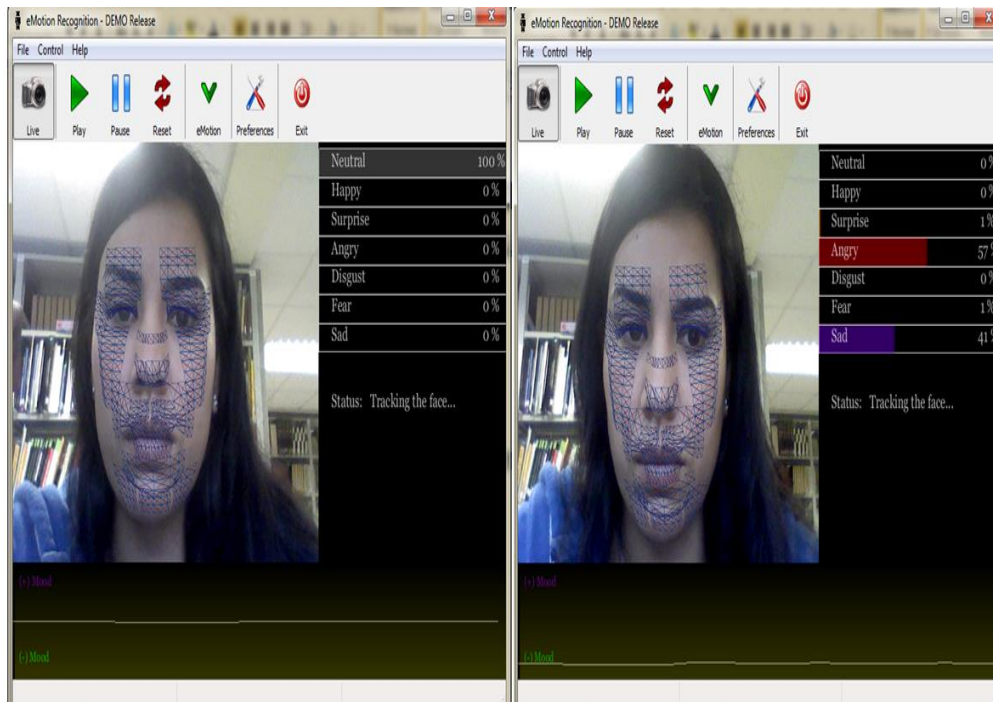


FIGURA 1.12 Resultados de la Polaridad de la imagen en tiempo real con EmotionRecognition.

Se analizó dos imágenes en tiempo real y se obtuvo los siguientes resultados:

- ✓ En la primera imagen se determinó un estado Neutral igual a 100%.
- ✓ En la segunda imagen se obtiene tres resultados: Angry 57%, Sad 41%, Surprise 1%, es decir, se obtiene una polaridad negativa.

1.14 Selección de Técnicas y Herramientas

Mediante la validación de varias técnicas y herramientas de extracción de emociones se ha tomado en consideración las siguientes:

- La herramienta seleccionada para el análisis de sentimientos en texto es **Sentiment AnalysisNLTK** por las ventajas que presenta, entre ellas: mantiene una licencia OpenSource, trabaja con 10 idiomas incluido el español que es necesario para este proyecto, además de trabajar con 50 corpus lingüísticos, extrae información de diferentes formatos de archivos, permite el procesamiento de contenidos: tokenización, extracción de raíces (lematización), eliminación de stopwords, corrección de textos (expresiones regulares, caracteres repetidos).

- Luego de la comparación realizada entre las herramientas NViso y EmotionRecognition, se concluye que el software más óptimo para el reconocimiento facial de emociones es EmotionRecognition, porque ofrece un SDK consumible mediante un API, que se adaptaría al desarrollo del agente de reconocimiento de emociones en imágenes, presente en la propuesta de Arquitectura.

1.15 Trabajos similares y Arquitecturas propuestas

✓ Análisis de opinión como un sistema multiagente distribuido

El proyecto realizado por (Kogan & Roger, 2011) pertenecientes al Grupo de Investigación en Lenguajes e Inteligencia Artificial, Facultad de Informática de la Universidad Nacional del Comahue, es parte del proyecto de investigación *Sistemas Multiagentes en Ambientes Dinámicos: Planificación, Razonamiento y Tecnologías del Lenguaje Natural*, y tiene como objetivo el desarrollo de conocimiento especializado en el área de Inteligencia Artificial Distribuida, estudiando técnicas de representación del conocimiento y razonamiento, en conjunto con métodos de planificación y tecnologías de lenguaje natural aplicadas al desarrollo de sistemas multiagentes, centrada en el desarrollo de aplicaciones destinadas al estudio y seguimiento de la opinión pública sobre un tema específico, cuyo principal objetivo es desarrollar una herramienta que nos permita realizar este proceso de manera automática. A continuación se presenta la Arquitectura básica del sistema multiagente para el análisis de opinión.

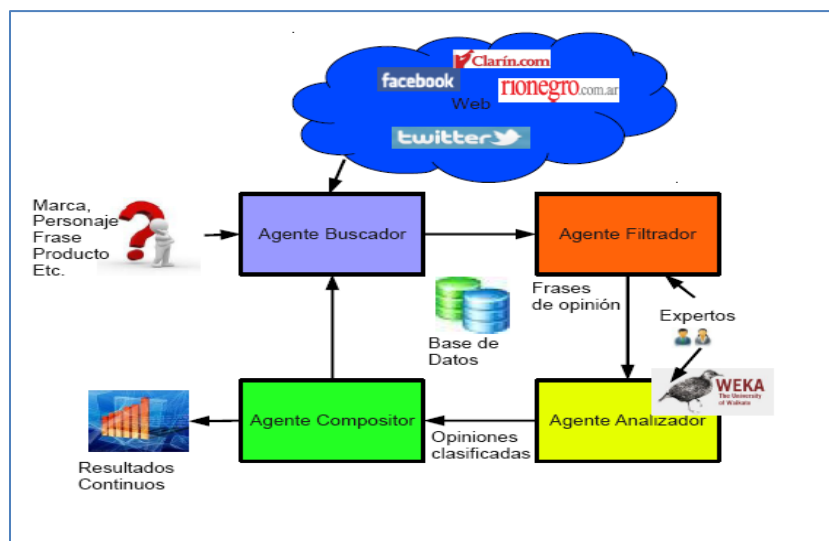


FIGURA 1.13 Arquitectura de Agentes para el análisis de opinión
Fuente:(Kogan & Roger, 2011).

Se puede observar que la arquitectura propuesta contiene cuatro agentes principales y son: Agente Buscador, Agente Filtrador, Agente Analizador y Agente Compositor.

Agente Buscador, se encarga de analizar las entradas, busca sobre la web y almacena lo buscado en una base de datos. Produce dos procesos, primero realiza la desambiguación de las entradas en caso de ser necesarias, el segundo trata también con la consulta pero en sentido de producir la relajación de la entrada. El proceso de búsqueda es realizado en: microblogs conocidos, comentarios de noticias de diarios, en la web a través de buscadores; toda esta información se almacena en una base de datos.

Agente Filtrador, aquí se descartan todos los datos del corpus que no sirven (entradas duplicadas, y entradas que no demuestran sentimientos).

Agente Analizador, clasifica las entradas del corpus. Es el encargado de realizar un proceso de entrenamiento sobre análisis de sentimientos, esta tarea es realizada con la herramienta WEKA, y el clasificador SVM (Support Vector Machine) inicialmente utilizado.

Agente Compositor, compone los resultados obtenidos por el agente analizador en un tiempo determinado. El factor tiempo y los resultados son los más importantes a analizar. Los resultados obtenidos pueden modificar el comportamiento del agente buscador antes de iniciar el nuevo ciclo.

✓ **Una Arquitectura Cognitiva Multinivel para Agentes con Comportamiento Influido por Características Individuales y Emociones, Propias y de Otros Agentes**

Esta Arquitectura fue propuesta por (Imbert, 2005) en su tesis doctoral, la misma que propone una arquitectura de carácter emocional, de vocación genérica, para el módulo cognitivo de los agentes denominada *Cognitiva*, que se identifica como un proceso continuo de *percepción-cognición-acción*, y distingue explícitamente o implícitamente tres módulos directamente relacionados con esos procesos: *sensores* para percibir el estado del entorno y sus cambios, *módulo cognitivo* para interpretarlo y adoptar las decisiones y *actuadores* para ejecutar las acciones decididas.

El modelo Arquitectónico más indicado para el tipo de agente de interés es una Arquitectura Híbrida, que combina la capacidad reactiva para proporcionar respuestas inmediatas, con las capacidades deliberativas para la elaboración de planes complejos. Dichas arquitecturas

se estructuran internamente por capas, el número de capas y su distribución física va en relación al propósito del sistema.

En la arquitectura propuesta se distingue tres áreas diferenciadas: sensores, módulo cognitivo y actuadores. En lo referente al módulo cognitivo, la arquitectura presenta una distribución en tres capas:

Capa reactiva, proporciona respuestas inmediatas según la variación del entorno.

Capa deliberativa, analiza la situación actual frente a las metas e intereses del agente, desde el punto de vista de sus capacidades personales para estructurar posibles soluciones en forma de planes.

Capa social, analiza la situación actual frente a las metas e intereses del agente, desde el punto de vista de la interacción con otros agentes del sistema, para así poder estructurar soluciones en forma de planes.

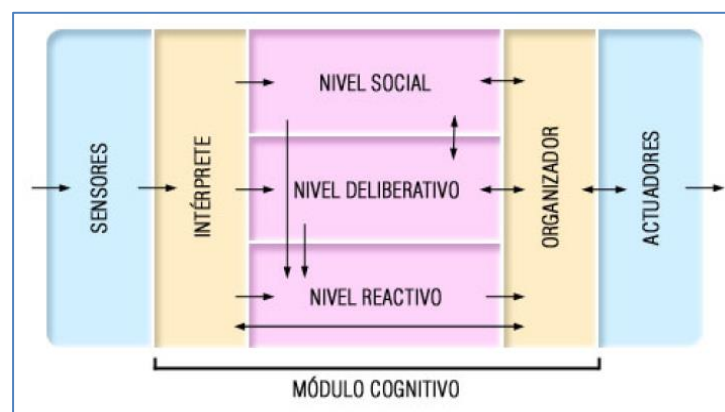


FIGURA 1.14 Arquitectura general del Agente
Fuente:(Imbert, 2005)

(Imbert, 2005) Manifiesta que la distribución de estas capas, con respecto a cómo acceden a los datos de la entrada y cómo generan la salida, cataloga al módulo Cognitiva como horizontal, ya que todas ellas tienen la capacidad de percibir la entrada a partir de los sensores simultáneamente, y todas generan en paralelo sus propuestas de acción. La comunicación de dichas propuestas de acción no se realiza directamente a los actuadores, sino a un elemento intermedio denominado *organizador* que es común a las tres capas antes mencionadas, y se encarga de ordenar y secuenciar las acciones hacia los actuadores. Se podría argumentar que este elemento común debería haber formado parte directamente de los actuadores, pero las decisiones que debe adoptar hacen que se considere más consecuente que forme parte del módulo cognitivo. Con la finalidad de minimizar el acoplamiento entre los sensores y el módulo cognitivo se introduce un

componente interno denominado *intérprete*, que almacena las percepciones proporcionadas por los sensores a partir de sensaciones recibidas, las filtra y da el tratamiento más adecuado, es decir, el intérprete “traduce” al formato que maneje el módulo cognitivo, para luego hacerle llegar a los componentes adecuados de dicho módulo. Dichas percepciones traducidas sirven para indicar la hipótesis interpretativa inicial de lo percibido en el módulo Cognitiva, por esta razón consideran que el intérprete forme parte del propio módulo cognitivo, ya que su interacción con los demás elementos va más allá de ser una simple capa de interfaz. En la arquitectura propuesta se puede identificar los siguientes flujos básicos de comunicación:

- El *intérprete* recibe continuamente de los sensores, notificaciones de cambios de estado, eventos e información.
- Los niveles reactivo, deliberativo y social, que conforman el módulo cognitivo, reciben del intérprete solo la información de eventos y cambios de estado que les resulte de más interés.
- Los niveles social y deliberativo procesan esa entrada y de acuerdo a ciertos parámetros generan nuevas metas y proponen posibles planes para cumplirlas, cada nivel desde su perspectiva particular.
- De acuerdo con las metas trazadas, indican al nivel reactivo las pautas que debe seguir en su proceder, para que vaya acorde con dichas metas.
- El nivel reactivo luego de recibir la entrada procesada por el intérprete y las indicaciones recibidas, propone al organizador las acciones que considera como necesarias.
- El organizador recoge los planes y acciones propuestos y los estructura para que no haya conflictos entre ellos, y trata de optimizar el comportamiento del agente.
- Una vez que el organizador genera la agenda, éste es capaz de ir proporcionando las acciones seleccionadas en el orden pertinente a los actuadores.
- Finalmente, el organizador comunica al intérprete cuales son las expectativas que se tienen de acuerdo con las acciones que se van ejecutando. Mientras que el intérprete indica al organizador si las percepciones recibidas van de acuerdo a las expectativas manifestadas.
- La relación en los comportamientos del agente se va a basar en tres ejes fundamentales: el mantenimiento del conocimiento/creencias para justificar y sustentar su proceder, actualización automática de los valores de dichas estructuras de información mediante las relaciones precisas entre ellas (ad hoc), y procesar las necesidades y creencias del agente para originar las acciones pertinentes en cada nivel de la arquitectura.

✓ **Modelado Multiagente para Sistemas Emergentes y Auto-Organizados (MASOES)**

En este trabajo realizado por (Perozo, 2011) se define una arquitectura multiagente que permite modelar un sistema emergente y auto-organizado, a través de un grupo de agentes (homogéneos y heterogéneos) descentralizados con diferentes comportamientos: reactivos, imitativo/cognitivo, permitiéndoles cambiar su comportamiento según su estado emocional para que se puedan adaptar dinámica a su entorno. Para conseguir esto, propone un modelo afectivo bidimensional (positiva y negativa), que nos permite conocer algunos aspectos de la interacción social multiagente. Además se detalla una metodología de uso para el modelado con MASOES, que explica como describir los elementos, relaciones y mecanismos a nivel individual y colectivo del grupo de agentes, permitiendo así el análisis del fenómeno emergente y auto-organizado sin necesidad de modelar matemáticamente el sistema.

En MASOES cada agente puede cambiar su comportamiento dinámicamente guiado por su estado emocional, para satisfacer dinámicamente los objetivos del sistema a través de la auto-organización de sus actividades.

La arquitectura propuesta por (Perozo, 2011) es la siguiente:

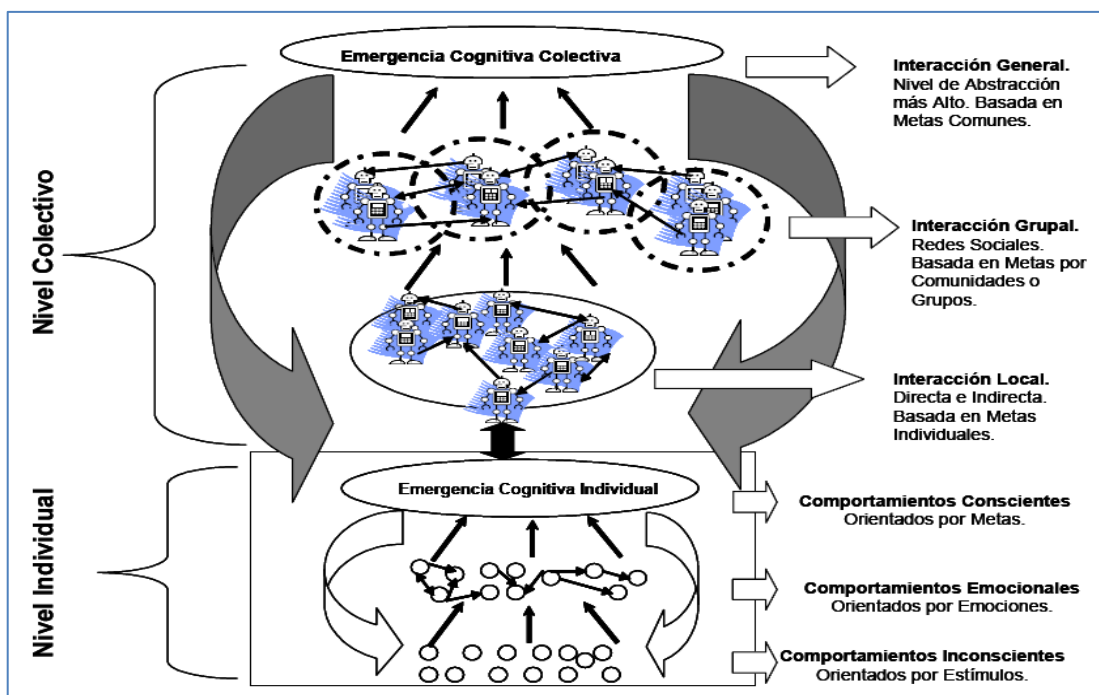


FIGURA 1.15 Arquitectura Multiagente para Sistemas Emergentes y Auto-Organizados
Fuente: (Perozo, 2011).

CAPÍTULO 2

Características de los estados afectivos que influyen en el aprendizaje

2.1 Introducción

El análisis de texto surge ante la necesidad de conocer, comprender y descifrar la carga emotiva presente en mails, chats, comentarios.

Por esta razón, en este capítulo se presentan algunos temas que son necesarios para comprender mejor el significado de los mensajes, entre ellos están el uso de emoticones, así como alteraciones lingüísticas, exageración de signos de puntuación, onomatopeyas y acrónimos, todos estos presentes en conversaciones.

A partir de estos conceptos planteamos escenarios para determinar el tipo de emoción presente en el texto obtenido de la Red Social del EVA, y determinar qué tipo de alteraciones presenta.

2.2 Análisis de Texto

Para realizar el análisis de texto es necesario tener en cuenta ciertos aspectos que influyen al momento de determinar su carga emocional, es por ello que se han considerado los siguientes puntos que se detallan a continuación.

2.2.1 Emoticones

La palabra “*emoticons*” (término en inglés) es un neologismo “emotion” (emoción) e “icon” (ícono, imagen), son un conjunto de símbolos que fueron creados para expresar emociones, actitudes o situaciones individuales. Los emoticones intentan representar emociones y completar el sentido de las oraciones, además de cumplir con una función informativa (Hernández & Miraflores, 2007).

Tabla 2.1 Emoticones en caracteres ASCII

Emoticones de uso frecuente	
:-) :)	Alegría
:(:(:-c	Tristeza
;-) ;)	Complicidad, humor, ironía, coqueteo

:-{	Miedo
[]	Abrazos
:-* **	Besos
:-S :-/	Ansiedad, preocupación, duda
:-o :-O :-@	Sorpresa, confusión, gritando
:- :- ~-:-[Disgustado, enfadado, ira
:-l :-,	Indiferente
:-P	Broma, ironía, confianza

Fuente:Adaptación de (López, 2009)

2.2.2 Alteraciones lingüísticas

Las aplicaciones más utilizadas para la comunicación en internet se encuentran llenas de errores ortográficos y tipográficos, abreviaciones, contracciones, usos indebidos de reglas gramaticales y puntuación, onomatopeyas y acrónimos. A continuación se describen las alteraciones lingüísticas más representativas utilizadas por los usuarios de Internet (independientemente de la herramienta utilizada) (Etchevers G, 2006):

1. Sustitución de letras consonantes complejas y morfemas (ch, ll, que) por grafemas simples (x,y,k).
2. Abreviación de morfemas (que=k, q; por=x; porque=pq, xq)
3. Contracción de palabras eliminando consonantes o sílabas (nada=na; encontrado=encontrado; mucho=mxo)
4. Uso de mayúsculas (¡NO ENTIENDO!)
5. Repetición de grafemas (me encaaaaaaanta; síííí)

Entre las justificaciones para esta incorrecta forma de escribir tenemos: priorizar la rapidez de escritura, falta de experiencia en la manipulación del teclado o botones, la intención

consciente de reproducir la fonética, la motivación por incorporar coloquialismos, sentimiento de inclusión, ahorro de letras.

De las alteraciones lingüísticas mencionadas anteriormente, las más relacionadas con el aspecto emocional son: el uso de Mayúsculas y repetición de vocablos.

El uso voluntario de MAYÚSCULAS es para que el texto logre transmitir figuradamente información paraverbal emocional que se expone generalmente de forma natural en una conversación cara a cara. Cuando se escribe con mayúsculas es para indicar “énfasis” o “reforzamiento”, ya que las palabras escritas en mayúscula causan más impacto que el resto del texto y significan un aumento en el tono de la conversación.

Las repeticiones o arrastres de grafemas reflejan textualmente el parámetro de extensión de los cualificadores vocales del paralenguaje, cuyo objetivo es igual que el uso de las mayúsculas, “destacar”, “recalcar” y “enfaticar” claramente una palabra. Cuando se lee un texto escrito en este formato se puede imaginar las inflexiones en el tono de voz de la persona que escribe.

2.2.3 Exageración de signos de puntuación

El uso desmesurado de los signos ortográficos o de puntuación forma parte de los componentes de la comunicación emocional virtual. Con el gran crecimiento de usuarios de internet es frecuente encontrar signos de puntuación como: puntos (.), comas (,), exclamaciones (!), interrogaciones (?), comillas (“”), puntos suspensivos “...”, que son utilizados sin regirse a ningún tipo de regla gramatical, simplemente que se adapte al usuario.

Las exageraciones en la escritura o repeticiones de los signos de exclamación permiten acrecentar el énfasis emocional de lo expresado, ya sea positivo o negativo. En el caso de los signos de interrogaciones son usados para mostrar una pregunta, sorpresa, duda o desconcierto más allá de lo normal.

Los signos de puntuación por lo general se encuentran al final de una frase y desaparecen al inicio de una frase, esto se debe al hecho de ahorrar tiempo al momento de escribir. Los puntos suspensivos son otros signos que llaman la atención por el desmedido uso que se hace en Internet (chat, mensajería, foros, red social, etc), implica suspender momentáneamente lo que se dice y tiende a utilizarse como separador de ideas y

representa pausas y silencios en una conversación presencial. Los puntos suspensivos en la escritura denotan un silencio o pausa por parte del usuario (Etchevers, 2008).

2.2.4 Onomatopeyas

Las nuevas formas de comunicación y los lenguajes utilizados para comunicarse mediante un ordenador han incorporado un uso fonético textual simulado “onomatopéyico” de algunas de las reacciones típicas de las emociones básicas universales, así como de muchos otros sonidos capaces de ser representados mediante vocablos escritos.

El diccionario de la Real Academia Española (DRAE) define una onomatopeya como una “imitación del sonido de una cosa en el vocablo que se forma para significarla”, por esta razón a falta de sonidos reales los usuarios de Internet tienen que representar y traducir de alguna manera el gran repertorio del paralenguaje con el que se cuenta en la realidad para expresar la mayoría de los estados emocionales. Las representaciones verbales dependen de las características como de las preferencias de cada persona, por ejemplo: para representar un estado de *sorpresa* se puede utilizar el término “Oh!!”, de *indecisión* “mmm, uhmm, humm”, *simpatía* o buen humor “jaja”

El uso de las onomatopeyas por parte de los usuarios de Internet puede interpretarse como la representación de personas creativas, espontáneas o imaginativas.

2.2.5 Acrónimos

Un acrónimo es la palabra que se forma utilizando únicamente la inicial de otras palabras, los acrónimos tienen un rol importante y de uso generalizado en las formas de comunicación dentro de este tipo de medios, porque crean nuevos acrónimos exclusivos para expresar frases comunes o representar acciones exactas con mayor rapidez y precisión al momento de escribir. Cabe recalcar que los acrónimos son más utilizados en chats o mensajería instantánea.

2.2.6 Diccionario Raíz

Es conveniente crear un diccionario de tipo raíz debido a la gran cantidad de conjugaciones que se puede derivar de una palabra. Para la construcción de este diccionario se tomó las

palabras sinónimas³ de la emoción principal y también se incluye los emoticones que los caracterizan.

Tabla 2.2 Palabras alineadas a emociones de tipo raíz

Simpatía					
Acepta	aprecio	consanguinidad	entusiasmo	instinto	resignación
adhesión	ardor	consenso	espíritu	interés	reverencia
adora	armonía	consentimiento	estima	jovialidad	superficialidad
afán	arrebato	contento	euforia	júbilo	ad
afecto	atractivo	coqueteo	exaltación	juicio	vehemencia
afectuoso	avenencia	cordialidad	familiaridad	ligereza	:)
aficionado	avidez	correlación	fanatismo	lindo	^_^
afinidad	belleza	cortés	fascina	lucido	:)
agradece	bondad	cortesés	felicidad	mágico	;D
alborozo	caricia	cortesía	frenesí	maravilloso	:D
algazara	cariño	delicia	frivolidad	mimo	=D
amable	cercanía	delirio	gozo	paciencia	=)
amistad	coincidencia	deslumbra	gracia	parentesco	:)
amor	compatible	devoción	halago	pasión	:>
analogía	compenetra	dicha	hermoso	persuasión	:>
ánimo	comprensión	emoción	humanidad	placer	:>
apasionamiento	concomitan	empeño	ilusión	predisposición	:D
apego	concordar	enardecimiento	inclinación	ratificación	;)
apoyo	conformidad	encanto	ingenio	regocijo	X-D
Ansiedad/Preocupación					
abatido	codicia	Desolado	incomodo	nostalgia	sufrir
afán	congojado	desvelo	indeciso	padecer	suplicio
afligido	conmocionado	difícil	indisposición	pasión	susto
agita	consternado	disgustado	inseguro	pena	temor
agobiado	contrariado	disturbio	insomnio	pesadumbre	tensión
agoniza	desaliento	dolor	intranquilidad	pesar	tormenta
ahogo	desasosiego	esfuerzo	intrigado	pésimo	tortura
alarmado	desazón	esquizofrenia	irrita	preocupa	tortura
albor	desconsolado	estrés	malestar	presión	tribula
amargado	desconsuelo	exaspera	manía	presionad	turba

³ Sinónimos tomados de: <http://www.wordreference.com>

2.2.7 StopWord List

El StopWord⁴ List, es una lista de palabras que carecen de significado, y no aportan valor alguno al momento de realizar el procesamiento de texto, por tal razón, se crea una lista de estas palabras para que se excluyan del texto que se está analizando.

Tabla 2.3 StopWord List

StopWord List					
A	dado	este	más	podrán	solas
actualmente	dan	éste	mayor	podría	solo
adelante	dar	esto	me	podrían	sólo
además	de	estos	mediante	poner	solos
afirmó	debe	éstos	mejor	por	son
agregó	deben	estuvo	mencionó	porque	su
ahí	debido	ex	menos	posible	sus
ahora	decir	existe	mi	primer	tal
al	dejó	existen	mientras	primera	también
algo	del	explicó	misma	primero	tan
algún	demás	expresó	mismas	primeros	tanto
alguna	dentro	fin	mismo	principalmente	tendrá
algunas	desde	fue	mismos	propia	tendrán
alguno	después	fuera	momento	propias	tenemos
algunos	dice	fueron	mucha	propio	tener
alrededor	dicen	gran	muchas	propios	tenga
ambos	dicho	grandes	mucho	próximo	tenía
ante	dieron	ha	muchos	próximos	tenido
anterior	diferente	haber	muy	pudo	tercera
antes	diferentes	había	nada	pueda	tiene
añadió	dijeron	habían	nadie	puede	tienen
apenas	dijo	habrá	ni	pueden	toda
aproximadamente	dio	hace	ningún	pues	todas
aquí	donde	hacen	ninguna	que	todavía
aseguró	dos	hacer	ningunas	qué	todo
así	durante	hacerlo	ninguno	queremos	todos
aún	e	hacia	ningunos	quien	total
aunque	ejemplo	haciendo	nos	quienes	tras
ayer	el	han	nosotras	quiere	trata

⁴ StopWord tomado de: <http://members.unine.ch/jacques.savoy/clef/spanishSmart.txt>

bajo	él	hasta	nosotros	realizado	través
bien	ella	hay	nuestra	realizar	tres
buen	ellas	haya	nuestras	realizó	tuvo
buena	ello	he	nuestro	respecto	última
buenas	ellos	hecho	nuestros	se	últimas
bueno	embargo	hemos	nueva	sea	último
buenos	en	hicieron	nuevas	sean	últimos
cada	encuentra	hizo	nuevo	según	un
casi	entonces	hoy	nuevos	segunda	una
cerca	entre	hubo	o	segundo	unas
cierto	era	igual	ocho	seis	uno
cinco	eran	incluso	otra	señaló	unos
comentó	es	indicó	otras	ser	usted
como	esa	informó	otro	será	va
cómo	esas	junto	otros	serán	vamos
conocer	ese	la	para	sería	van
considera	eso	lado	parece	si	varias
consideró	esos	las	parte	sí	varios
contra	esta	le	partir	sido	veces
cosas	está	les	pasada	siempre	ver
creo	ésta	llegó	pasado	siendo	vez
cual	estaba	lleva	pero	siete	y
cuales	estaban	llevar	pesar	sigue	ya
cualquier	estamos	lo	poca	siguiente	yo
cuando	están	los	pocas	sin	
cuanto	estar	luego	poco	sino	
cuatro	estará	lugar	pocos	sobre	
cuenta	estas	manera	podemos	sola	
da	éstas	manifestó	podrá	solamente	

2.3 Análisis de los diálogos que presentan estados emocionales

Para poder determinar las emociones que influyen directamente en el aprendizaje de los estudiantes se ha considerado tomar una muestra de 10 mensajes, obtenidos de la Red Social del EVA, pertenecientes a estudiantes del 10mo ciclo de la Carrera de Informática de Modalidad Abierta y a Distancia, con los cuales se pretende plantear un escenario de análisis para realizar el procesamiento de texto y determinar su estado emocional, partiendo

del diccionario raíz se obtiene un conjunto de palabras claves y las características asociadas a cada estado emocional.

2.3.1 Escenario: Extracción de palabras que determinan el tipo de emoción en texto, a través de la Red Social del EVA

Para poder extraer las palabras clave y determinar el tipo de emoción se ha tomado como referente los siguientes 10 mensajes:

Texto analizado 1:

Buenos días Ing. Por **favor** nos **podría** indicar las preguntas de Autoevaluación de Repaso las subimos al EVA, y como **podemos** subirlas. Mil **gracias**

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Sin exageración de signos de puntuación.
- ✓ Presenta dos acrónimos: Ing, EVA.
- ✓ Las palabras claves encontradas en el mensaje son: favor, podría, podemos, gracias.
- ✓ Se manifiestan 3 tipos de emociones: Ansiedad/Preocupación (-), Frustración (-) y Simpatía (+).

Texto analizado 2:

Buenas noches ingeniera seria tan amable de darnos un poco mas de **tiempo** para entregar los mapas conceptuales, son 2 y solo tenemos una semana...Espero **puedaayudarnos..Gracias**

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Contiene exageración de signos de puntuación (...).
- ✓ No presenta acrónimos.
- ✓ Las palabras claves encontradas en el mensaje son: tiempo, pueda, ayudarnos, Gracias.
- ✓ Se manifiestan 3 tipos de emociones: Ansiedad/Preocupación (-), Frustración (-) y Simpatía (+).

Texto analizado 3:

Por cierto y la nota del 1B???. Como se que estoy bien en su materia???

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Contiene exageración de signos de puntuación (???)
- ✓ No presenta acrónimos.
- ✓ Las palabras claves encontradas en el mensaje son: ???.
- ✓ Se manifiesta 1 tipo de emoción: Confusión (-).

Texto analizado 4:

Estimada profesora, **podría** indicarme cual es el contexto del poster? la empresa que analizamos o la situación de las TICs en la ciudad?

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.

- ✓ No presenta alteraciones lingüísticas.
- ✓ Sin exageración de signos de puntuación.
- ✓ Presenta un acrónimo: TICs.
- ✓ Las palabras claves encontradas en el mensaje son: Estimada, podría, ?, ?.
- ✓ Se manifiestan 3 tipos de emociones: Confusión (-), Frustración (-) y Simpatía (+).

Texto analizado 5:

Ingeniera nos **podría** decir a donde subimos el informe....o en donde esta el enlace???

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Contiene exageración de signos de puntuación (...) y (???)
- ✓ No presenta acrónimos.
- ✓ Las palabras claves encontradas en el mensaje son: podría, ???.
- ✓ Se manifiestan 2 tipos de emociones: Confusión (-) y Frustración (-).

Texto analizado 6:

Saludos Ing. el segundo trabajo ud. publico que lo **podríamos** entregar hasta el 6 de agosto. Según el anuncio salen a vacaciones el 13.**Podemos** entregar hasta esa fecha?

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Sin exageración de signos de puntuación.
- ✓ Presenta dos acrónimos: Ing, ud.

- ✓ Las palabras claves encontradas en el mensaje son: Saludos, podríamos, Podemos, ?.
- ✓ Se manifiestan 3 tipos de emociones: Confusión (-), Frustración (-) y Simpatía (+).

Texto analizado 7:

No puedo insertar el video... no se que pasa, alguien puede **ayudarme?**

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Contiene exageración de signos de puntuación (...).
- ✓ No presenta acrónimos.
- ✓ Las palabras claves encontradas en el mensaje son: No puedo, ayudarme, ?.
- ✓ Se manifiestan 3 tipos de emociones: Ansiedad/Preocupación (-), Confusión (-) y Frustración (-).

Texto analizado 8:

Buenas tardes Ing. la empresa tambien **puede** ser de caracter comercial como un autoservicio o no?.**Gracias**

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Sin exageración de signos de puntuación.
- ✓ Presenta un acrónimo: Ing.
- ✓ Las palabras claves encontradas en el mensaje son: puede, ?, Gracias.
- ✓ Se manifiestan 3 tipos de emociones: Confusión (-), Frustración (-) y Simpatía (+).

Texto analizado 9:

Estimada profesora si es posible que nos oriente con metodología UML?

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Sin exageración de signos de puntuación.
- ✓ Presenta un acrónimo: UML.
- ✓ Las palabras claves encontradas en el mensaje son: Estimada, ?.
- ✓ Se manifiestan 2 tipos de emociones: Confusión (-) y Simpatía (+).

Texto analizado 10:

Buenas tardes Ingeniera. ¿Cuándo **podremos** saber las **notas** del primer bimestres?

Obteniendo los siguientes resultados:

- ✓ No contiene emoticones.
- ✓ No presenta alteraciones lingüísticas.
- ✓ Sin exageración de signos de puntuación.
- ✓ No presenta acrónimos.
- ✓ Las palabras claves encontradas en el mensaje son: ¿, podremos, notas, ?.
- ✓ Se manifiestan 3 tipos de emociones: Ansiedad/Preocupación (-), Confusión (-) y Frustración (-).

2.3.2 Palabras que determinan el estado emocional

Entre las palabras que nos permiten determinar el estado emocional de los estudiantes tenemos la siguiente clasificación obtenida del análisis textual de los mensajes de la Red Social, considerando el diccionario raíz y el tipo de emoción.

Tabla 2.4 Palabras obtenidas del análisis textual

Tipo de emoción	Simpatía	Ansiedad/Preocupación	Frustración	Confusión
Palabras clave	Estimada Cordial Gracias Saludos	Ayudarnos Ayudarme Error Examen Favor Notas	Podría Podemos Pueda Podríamos puedo puede podremos No Tiempo	¿ ? ???
Simbología	☺ :D ;) :* :P ^_^	☹ ☹ =S	:' >.< >\ x(?:-(:-/ :-\ :-O

2.3.3 Características de los estados emocionales

Entre las características que podemos destacar de los estados emocionales de los estudiantes tenemos:

Simpatía:

- ✓ Expresión de alegría y optimismo.
- ✓ Refleja dinamismo en sus palabras.
- ✓ Expresa agradecimiento por la ayuda recibida.
- ✓ Es muy amable y cordial con los demás.

Ansiedad/Preocupación

- ✓ Expresiones de tensión, inquietud, susto y miedo en relación con el aprendizaje (Rebollo, García, Barragán, Buzón, & Vega, 2008).

- ✓ Bajo rendimiento.
- ✓ Aproximación de las evaluaciones, proyectos, tareas.
- ✓ Baja productividad en las tareas.

Frustración

- ✓ El estudiante puede frustrarse por no tener estrategias y destrezas adecuadas para la formación online (Borges, 2005).
- ✓ No dispone de tiempo suficiente para cumplir con sus actividades académicas.
- ✓ El componente académico no cumple totalmente a sus objetivos.
- ✓ Desconoce la ayuda que ofrece el docente a través del entorno virtual.
- ✓ No cumple sus objetivos.

Confusión

- ✓ Muchas interrogantes, pocas respuestas.
- ✓ Se plantea varias alternativas y no sabe cuál elegir.
- ✓ Estudiante activo, pero no sabe que ruta seguir y pregunta por una solución.
- ✓ No tiene sus ideas claras.

CAPÍTULO 3

APLICACIÓN DE LA METODOLOGÍA SELECCIONADA PARA EL DESARROLLO DE AGENTES DE EXTRACCIÓN DE EMOCIONES

3.1 Metodología seleccionada MAS-CommonKADS

Para el desarrollo de este proyecto se ha seleccionado la metodología MAS-CommonKADS (Iglesias, 1998), que se enfoca principalmente en la construcción de sistemas inteligentes, permite la integración de técnicas de ingeniería de conocimientos e ingeniería de software orientada a objetos y de protocolos, para una mejor comprensión de esta metodología se explica en el siguiente apartado una visión general de la metodología seleccionada.

✓ Modelos de MAS-CommonKADS

MAK-CommonKADS propone los siguientes modelos para el desarrollo de sistemas multiagentes (Iglesias, 1998):

- **Modelo de Agente (AM)**, se especifica las características de los agentes que conformaran nuestra arquitectura, y la descripción de sus funciones.
- **Modelo de Organización (OM)**, se detalla la organización de los agentes y su relación con su entorno.
- **Modelo de Tareas (TM)**, se describe la descomposición funcional del sistema, las actividades necesarias que debe realizar para cumplir con sus objetivos.
- **Modelo de Comunicación (CM)**, se modela las interacciones hombre-máquina, se descompone el diálogo entre el usuario y el sistema, indicando cuales son las transacciones que se realizan.
- **Modelo de Coordinación (CoM)**, se especifica las interacciones de los distintos agentes que conforman el sistema.
- **Modelo de Diseño (DM)**, se describe la arquitectura y diseño del sistema multiagente como paso previo a su implementación.

✓ Fases de desarrollo de MAS-CommonKADS

El modelo del ciclo de vida para el desarrollo de sistemas multiagente sigue las siguientes fases (Iglesias, 1998):

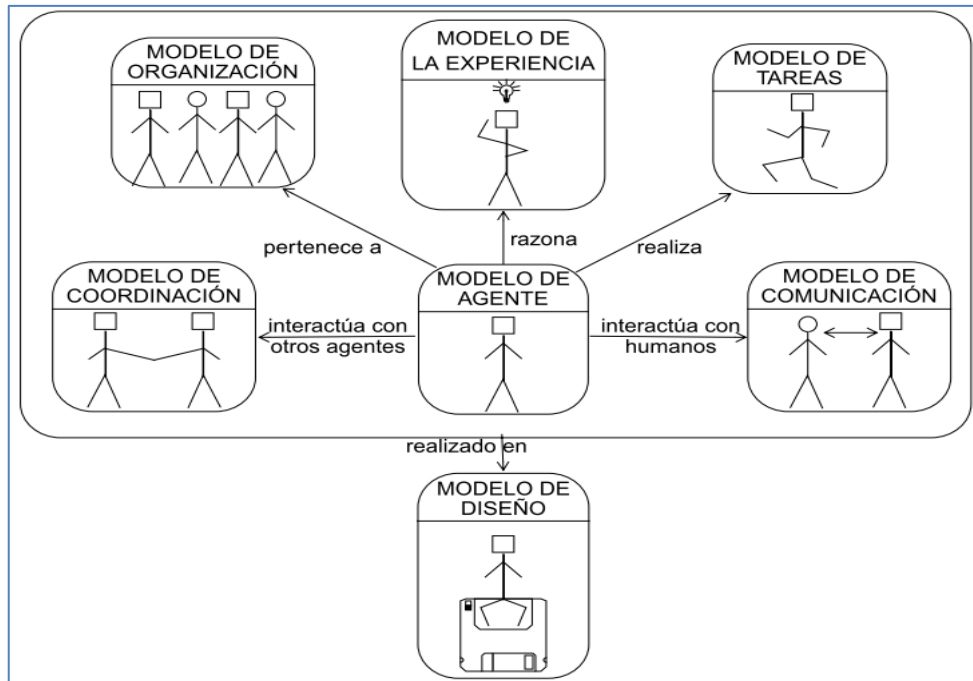


FIGURA 3.1 Modelos de MAS-CommonKADS
Fuente: (Iglesias, 1998)

- ✓ **Conceptuación**, tarea de elicitación⁵ para obtener una primera descripción del problema y la determinación de los casos de uso que pueden ayudar a entender los requisitos informales y a probar el sistema.
- ✓ **Análisis**, determinación de los requisitos de nuestro sistema partiendo del enunciado del problema. En esta fase se desarrollan los modelos de organización, tareas, agente, comunicación, coordinación y experiencia.
- ✓ **Diseño**, determinación de cómo los requisitos de la fase de análisis pueden ser logrados mediante el desarrollo del modelo de diseño. Se determinan las arquitecturas tanto de la red multiagente como de cada agente.
- ✓ **Codificación y prueba** de cada agente.
- ✓ **Integración**, el sistema completo es probado.
- ✓ **Operación y mantenimiento**.

El modelo de ciclo de vida propuesto para desarrollar estas tareas en la metodología es el modelo en espiral dirigido por riesgos, siguiendo la gestión de proyectos de CommonKADS. Tanto para proyectos pequeños como para el aprendizaje de la metodología se propone seguir un modelo de ciclo de vida basado en un modelo en cascada con reutilización (Iglesias, 1998).

⁵ Extracción o adquisición de conocimiento

✓ **Modelo de proceso software**

El paradigma de agentes tiene también como objetivo la interoperabilidad del software y parece apropiado hacer énfasis en el empleo de reutilización de componentes, dado que los agentes (Iglesias, 1998):

- Facilitan su ensamblaje a través del empleo de un lenguaje común de comunicación entre agentes, que está pensado para hacer posible su interoperabilidad
- Pueden ser entidades relativamente complejas, tienen una arquitectura software y hardware y tienen conocimiento de determinados dominios, permitiendo su reutilización
- Se caracterizan por su autonomía y pueden decidir con quién se “ensamblan” dinámicamente, si hay varios componentes posibles

La introducción a la reutilización de componentes puede realizarse bajo dos perspectivas(Iglesias, 1998):

- Cuando se tiene el análisis y los componentes necesarios ya identificados, se busca en una biblioteca de componentes. Si existen se adaptan, caso contrario se desarrollan.
- El proceso es dirigido por reutilización, primero se buscan los componentes y luego se realiza el diseño.

En un sistema multiagente se puede identificar varios tipos de componentes: el agente, las ontologías, los métodos de resolución de problemas, los servicios de un agente, sus objetivos y sus planes.

3.2 Aplicación de la Metodología

3.2.1 CONCEPTUALIZACIÓN

• Comprensión del Problema

Uno de los mayores problemas del aprendizaje en un entorno virtual, es la dificultad para detectar el estado emocional de los estudiantes, impidiendo brindarles retroalimentación para mejorar su motivación y su rendimiento académico.

• Ciclo de Desarrollo de los Casos de Uso

Los casos de uso describen una secuencia de interacciones que se realiza entre un sistema y sus actores. Los actores son externos al sistema.

✓ Identificación y Descripción de los Actores

Actores Activos:

- **Estudiante:** interactúa con el docente a través del entorno virtual de aprendizaje (EVA) mediante foros, mensajes, comentarios, videos.
- **Docente:** interactúa con el estudiante a través del entorno virtual de aprendizaje (EVA) mediante tareas, tutoriales, foros, mensajes, comentarios, videos, presentaciones. Este actor maneja más privilegios que el estudiante.

Actores Pasivos:

- **Base de Datos:** es el componente dentro del sistema encargado de almacenar los datos obtenidos de la interacción entre los actores.

✓ Identificación y Descripción de los Casos de Uso

Se identifican los siguientes casos de uso para los siguientes actores:

- **Docente/Estudiante:** tiene 4 casos de uso en común:
 - Iniciar/cerrar sesión en el entorno virtual de aprendizaje (EVA).
 - Actualizar datos.
 - Administrar contenidos (visualizar recursos, cargar archivos).
 - Responder Mensajes/foros/comentarios.
 - El **Docente** cuenta con un caso de uso adicional que difiere del estudiante:
 - Plantear tareas (tareas, foros, evaluaciones).
- ✓ Para el caso de *iniciar/cerrar sesión*, el docente/estudiante tiene que ingresar su ID y password para ingresar al Entorno Virtual de Aprendizaje (EVA), y cuando termina su interacción con el entorno procede a cerrar sesión.
- ✓ En el caso de *actualizar datos*, el docente/estudiante tiene la posibilidad de editar su información disponible en el EVA.
- ✓ *Administrar contenidos*, hace referencia a que el docente / estudiante puede visualizar y seleccionar el componente académico para tener acceso a los recursos/archivos disponibles, selecciona el recurso que desea visualizar/ descargar, en el caso de cargar un archivo se selecciona y se lo carga en el sistema.

- ✓ *Responder mensajes/foros/comentarios*, el docente/estudiante puede responder mensajes, foros o comentarios, según su necesidad, ingresa su respuesta y el sistema la registra. En el caso del estudiante puede enviar mensajes al docente en caso de tener alguna inquietud para obtener recursos adicionales.
- ✓ *Plantear tareas*, el docente puede agregar, modificar o eliminar los recursos educativos plantear foros y formular evaluaciones, para medir el nivel de conocimiento del estudiante y ofrecerle una guía o un fundamento teórico para una mejor comprensión del componente.

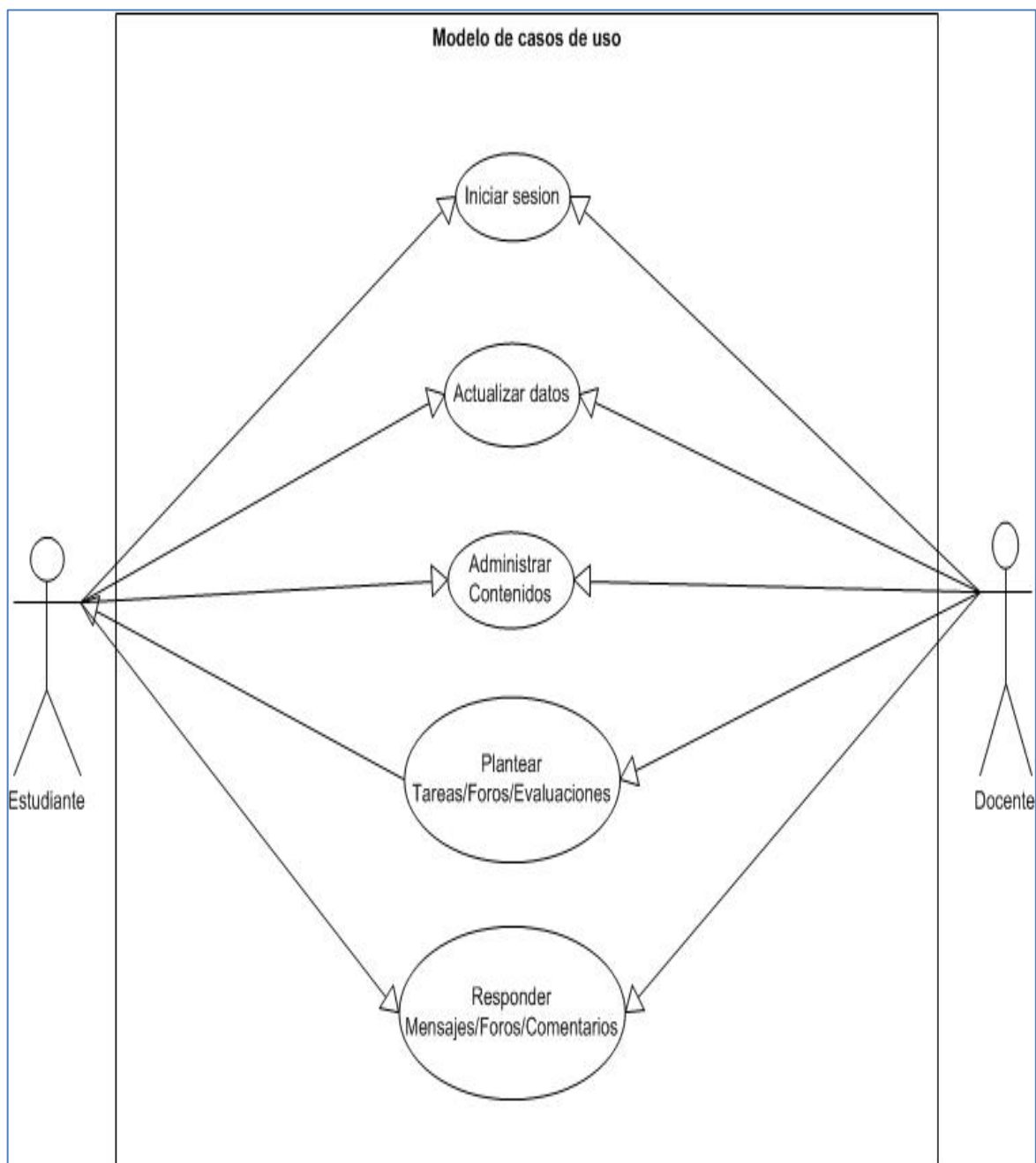


FIGURA 3.2 Casos de Uso para los Actores Activos

➤ Descripción de los Casos de Uso

Número:	01	
Nombre:	Iniciar sesión	
Actores:	Docente, Estudiante	
Descripción:	Su función es permitir que el usuario ingrese al sistema mediante la comprobación de su ID y password.	
Precondiciones:	El sistema debe estar habilitado para iniciar sesión	
Flujo Normal:	Actor:	Sistema:
	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de iniciar sesión. 2. Ingresa su ID y password 4. El servidor verifica si los datos ingresados por el usuario son válidos FA 2 	<ol style="list-style-type: none"> 3. Establece una conexión con el servidor. FA 1 5. Carga los datos del usuario, los componentes académicos. 6. Fin del Caso de Uso.
Flujo Alternativo:	<p>FA 2 Conexión fallida con el servidor Si la conexión con el Servidor falla el sistema mostrara un mensaje de conexión fallida o incorrecta</p> <p>FA 2 Datos ingresados no son válidos Ocurre cuando los datos de usuario ingresados son incorrectos, en ese caso el sistema desplegara un mensaje de datos incorrectos.</p>	
Excepciones:		

Número:	02	
Nombre:	Actualizar datos	
Actores:	Docente, Estudiante	
Descripción:	Su función es permitir la modificación del perfil de usuario.	
Precondiciones:	Que el usuario haya iniciado sesión	
Flujo Normal:	<p style="text-align: center;">Actor:</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción de Editar Información dentro del sistema. 3. El usuario ingresa los campos que desea actualizar. 	<p style="text-align: center;">Sistema:</p> <ol style="list-style-type: none"> 2. Se muestra una ventana con información sobre el usuario y con campos para ser reingresados. 4. Muestra un mensaje de actualización exitosa. 5. Fin del Caso de Uso.
Flujo Alternativo:		
Excepciones:		

Número:	03	
Nombre:	Administrar contenidos	
Actores:	Docente, Estudiante	
Descripción:	Su función es permitir que el docente/estudiante pueda interactuar con los contenidos de un componente académico.	
Precondiciones:	Que el docente/estudiante haya ingresado al sistema	
Flujo Normal:	<p style="text-align: center;">Actor:</p> <p>2. El docente / estudiante selecciona el componente académico.</p> <p>4. El docente / estudiante selecciona el recurso que desea visualizar/descargar. En el caso de cargar un archivo se selecciona y se lo carga en el sistema.</p>	<p style="text-align: center;">Sistema:</p> <p>1. Se muestra todos los componentes académicos a los que este suscrito el docente/estudiante.</p> <p>3. Muestra los recursos/ archivos disponibles en el componente académico.</p> <p>5. El sistema carga/descarga el recurso hacia o desde el componente respectivamente.</p> <p>6. Fin del Caso de Uso.</p>
Excepciones:		

Número:	04	
Nombre:	Plantear Tareas/Foros/Evaluaciones	
Actores:	Docente	
Descripción:	Su función es permitir que el docente formule tareas, foros, evaluaciones.	
Precondiciones:	El docente debe estar registrado en el sistema	
Flujo Normal:	Actor:	Sistema:
	<ol style="list-style-type: none"> 1. El docente selecciona el componente académico al que desea ingresar. 2. El docente selecciona si desea cargar tareas, foros o evaluaciones. 4. El docente ingresa la tarea, foro o evaluación. 	<ol style="list-style-type: none"> 3. El sistema carga los contenidos seleccionados. 5. El sistema registra la interacción del docente. 6. Fin del Caso de Uso.
Excepciones:		

Número:	05	
Nombre:	Responder Mensajes/Foros/Comentarios	
Actores:	Docente, Estudiante	
Descripción:	Su función es permitir que el docente/estudiante responda un mensaje, foro o comentario.	
Precondiciones:	El docente/estudiante debe estar registrado en el sistema.	
Flujo Normal:	Actor:	Sistema:
	<ol style="list-style-type: none"> 1. El docente / estudiante selecciona el componente académico al que desea ingresar. 2. El docente / estudiante selecciona si desea responder mensajes, foros o comentarios. 4. El docente / estudiante ingresa la respuesta. 	<ol style="list-style-type: none"> 3. El sistema carga los contenidos seleccionados. 5. El sistema registra la respuesta. 6. Fin del Caso de Uso.
Excepciones:		

3.2.2 Estructura de Agentes de extracción de emociones en Texto

➤ ANÁLISIS

Consiste en realizar un análisis de los componentes que formaran parte de la arquitectura, a partir de estos se puede realizar el diseño tanto físico y lógico, para la extracción de emociones en texto.

✓ Modelo de Agente

Especifica las características de un agente, se orienta esencialmente a los servicios.

▪ Identificación y descripción de los Agentes

Para poder determinar los agentes del sistema se consideró la diferencia entre las tareas y los roles de los actores.

- **Agentes Humanos**, son los actores activos identificados en la fase de conceptualización, y son: Docente y Estudiante
- **Agentes Software**, una vez analizadas las tareas, se determinó los siguientes agentes: Extracción, Segmentación, Lematización, Ontología, Polaridad, Organizador.

▪ Identificación de las características de los agentes

Para distinguir los agentes del sistema se considera la diferencia entre tareas y roles de los actores.

Tabla 3.1 Identificación del Agente Extracción

Agente	Extracción	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Conocimiento de las entradas de texto (mensajes) y su constante actualización.

Descripción	Agente software encargado de recolectar y almacenar todos los mensajes obtenidos de la interacción del estudiante con el docente a través del componente académico, se adapta a los cambios del entorno (constante actualización). Envía todos los datos recolectados a una base de datos (BD_Emociones), para su posterior procesamiento.
--------------------	--

Tabla 3.2 Identificación del Agente PLN

Agente	PLN	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Extrae información de la base de datos para realiza el análisis del texto, mediante la utilización de stopwords.
Descripción	Este agente esta encargado de extraer los datos que se almacena en la base (BD_Emociones), para realizar el proceso de eliminación de palabras irrelevantes, mediante el stopword	

Tabla 3.3 Identificación del Agente Polaridad

Agente	Polaridad	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Determina si el texto es positivo o negativo (polaridad de las palabras)
Descripción	Este agente tiene como entrada un archivo txt resultado del procesamiento del agente PLN para determina el tipo de emociones presentes en el texto analizado, mediante la utilización de un diccionario afectivo, palabras raíz, verbos, negaciones, intensificadores.	

Tabla 3.4 Identificación del Agente Ontología

Agente	Ontología	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Mediante un dominio obtenido de la Lematización se obtendrá tripletas para facilitar el intercambio de información.
Descripción	<p>El agente Ontología es la entidad encargada de establecer un esquema conceptual sobre uno o varios conjuntos de texto, que son resultado del agente Polaridad.</p> <p>Este agente es implementado con el fin de facilitar el intercambio y comunicación de datos, conceptos e información entre distintos sistemas y entidades.</p> <p>La forma de trabajo del mismo será seleccionar un dominio de los existentes dentro de la base de datos resultante del análisis de la Polaridad del texto.</p>	

Tabla 3.5 Identificación del Agente Organizador

Agente	Organizador	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Coordina el trabajo de los agentes para que trabajen secuencialmente.
Descripción	<p>El agente organizador es la entidad encargada de recolectar las interacciones de los otros agentes que interactúan entre si dentro del sistema para llevar a cabo el análisis de sentimientos.</p> <p>Se lo podría describir como el cerebro o centro de todo el sistema, debido a que se encarga de gestionar las tareas, recursos y procesos con los que trabajan los demás agentes.</p>	

Tabla 3.6 Identificación del Agente Interfaz

Agente	Interfaz	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Presenta la polaridad obtenida como resultado del procesamiento de texto.
Descripción	Tiene como entrada los resultados del análisis del texto realizado por el agente Polaridad, según la polaridad que se obtenga, el agente Interfaz envía un “alerta” al docente para que este pueda brindar retroalimentación al estudiante, mediante una interfaz gráfica.	

- **Distribución tareas-agentes**

Para visualizar de mejor manera cuáles son las tareas que deben realizar cada uno de los agentes que conforman el sistema, es conveniente presentar la Tabla 3.7 en la que tenemos a cada uno de los agentes que conforman el sistema y las tareas que deben desempeñar.

Tabla 3.7 Distribución tareas-agentes

	Obtener datos	Eliminación de palabras irrelevantes	Categorización de datos	Polaridad (Diccionario afectivo)	Conceptualización	Visualización de datos
Extracción	X					
PLN		X				
Polaridad			X	X		
Ontología					X	
Organizador	X	X		X	X	X
Interfaz						X
BD_Emociones	X	X		X	X	

- **Identificación y descripción de los Objetivos**

Para identificar los objetivos partimos de la asignación de tareas a cada agente, y tenemos:

Objetivo: Adaptarse al usuario

Parámetros-entrada

Entradas de texto como: mensajes, foros, comentarios.

Parámetros-salida

Fichero con el texto ingresado

Condición-activación

Al ingresar texto el estudiante

Condición- finalización

Cuando no existen nuevas entradas de texto

Descripción

El agente Extracción tiene como objetivo recolectar y almacenar todos los mensajes obtenidos de la interacción del estudiante con el docente.

Objetivo: Eliminación de palabras irrelevantes

Parámetros-entrada

Toma información de la base de datos obtenida del entorno (mensajes) por el agente Extracción

Parámetros-salida

Texto libre de palabras irrelevantes almacenado en un txt

Condición-activación

Extrae la información obtenida por el agente Extracción para procesarla

Condición- finalización

Que no exista información pendiente de procesar

Descripción

Este agente esta encargado de extraer los datos que se almacena en la base (BD_Emociones), realiza el proceso de eliminación de palabras irrelevantes, mediante la utilización de un stopwords list.

Objetivo: Determinar el estado emocional de los estudiantes

Parámetros-entrada

Texto sin palabras irrelevantes

Parámetros-salida

Polaridad de la información (+ o -) y tipo de emoción

Condición-activación

Toma la información proporcionada por el agente PLN

Condición- finalización

No tener información pendiente de procesar

Descripción

Este agente tiene como entrada las palabras resultantes del agente PLN para determina su polaridad (+ o -), mediante el uso de un diccionario afectivo, palabras de tipo raíz, verbos, negaciones e intensificadores.

Objetivo: Crear un soporte ontológico

Parámetros-entrada

Texto analizado

Parámetros-salida

Base de datos de tripletas

Condición-activación

Texto con Polaridad

Condición- finalización

Obtención de tripletas

Descripción

El agente Ontología es la entidad encargada de establecer un esquema conceptual sobre uno o varios conjuntos de texto, que son resultado del agente Polaridad.

Este agente es implementado con el fin de facilitar el intercambio y comunicación de datos, conceptos e información entre distintos sistemas y entidades.

Como resultante de este agente se obtendrá una base de datos de tripletas con las emociones encontradas.

Objetivo: Visualización de resultados

Parámetros-entrada

Información resultante del Agente Polaridad

Parámetros-salida

Interfaz gráfica con los resultados de la polaridad del texto y tipo de emoción

Condición-activación

Extrae la información obtenida por el agente Ontología

Condición- finalización

Que no exista información pendiente de presentar

Descripción

El agente Interfaz presenta mediante una interfaz gráfica los resultados obtenidos del análisis textual

Objetivo: Coordinar todas las interacciones entre los agentes

Parámetros-entrada

Información procesada de todos los agentes

Parámetros-salida

Información para que procesen los agentes en un orden específico

Condición-activación

Con el primer procesamiento de información que hace el agente de Extracción

Condición- finalización

Cuando el agente Interfaz presenta la información resultando del procesamiento de texto

Descripción

El agente organizador es la entidad encargada de recolectar información de los agentes que interactúan entre sí dentro del sistema para llevar a cabo el análisis de sentimientos.

✓ Modelo de Tareas

Describe la descomposición funcional de sistema, es decir, las tareas que los agentes pueden realizar.

Tarea Análisis de Sentimientos**Objetivo**

Obtener estados emocionales del estudiante

Descripción

En esta tarea se determina el estado emocional de los estudiantes en un curso virtual

Entrada

Mensajes, comentarios, foros (texto)

Salida

Polaridad de las Palabras

Precondición

Texto (mensajes, comentarios, foros) recibido

Supertarea

Procesamiento de texto 1

Subtareas

Obtener datos 1.1.1, Eliminación de palabras irrelevantes 1.1.2, Categorización de datos 1.1.3, Polaridad 1.1.4, Conceptualización 1.1.5, Visualización de datos 1.1.6.

Tipo de descomposición

Temporal/Funcional

Ingrediente Polaridad Emociones

Descripción

Polaridad de emociones implica analizar emociones en texto, para determinar su polaridad y determinar si el estudiante necesita retroalimentación en algunos temas del componente académico.

▪ Descomposición de tareas

A continuación se presenta la descomposición de las tareas en subtareas, mediante diagramas de Definición Integrada para Modelado de Funciones (IDEF). En la FIGURA 3.3 se describe la tarea Análisis de Sentimientos, para lo cual es necesario que el docente envíe los mensajes que desea procesar, luego se realiza la eliminación de las palabras irrelevantes, pasa a la tarea Polaridad para ser procesado y a partir de estos resultados se realiza la creación de conceptos y luego su visualización.

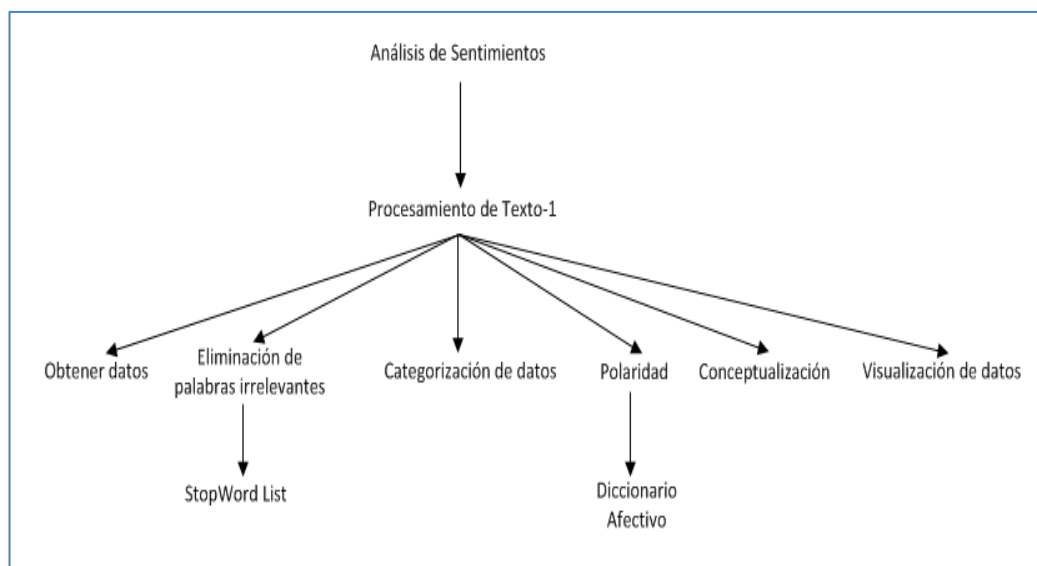


FIGURA 3.3 Descomposición de la Tarea T1: Análisis de Sentimientos

✓ **Modelo de Coordinación**

Nos permite profundizar en las interacciones de los distintos agentes que conforman el sistema.

▪ **Identificación de las Conversaciones**

Las distintas conversaciones se identifican utilizando la técnica de casos de uso internos. Luego se realiza una descripción de estas conversaciones mediante el uso de plantillas.

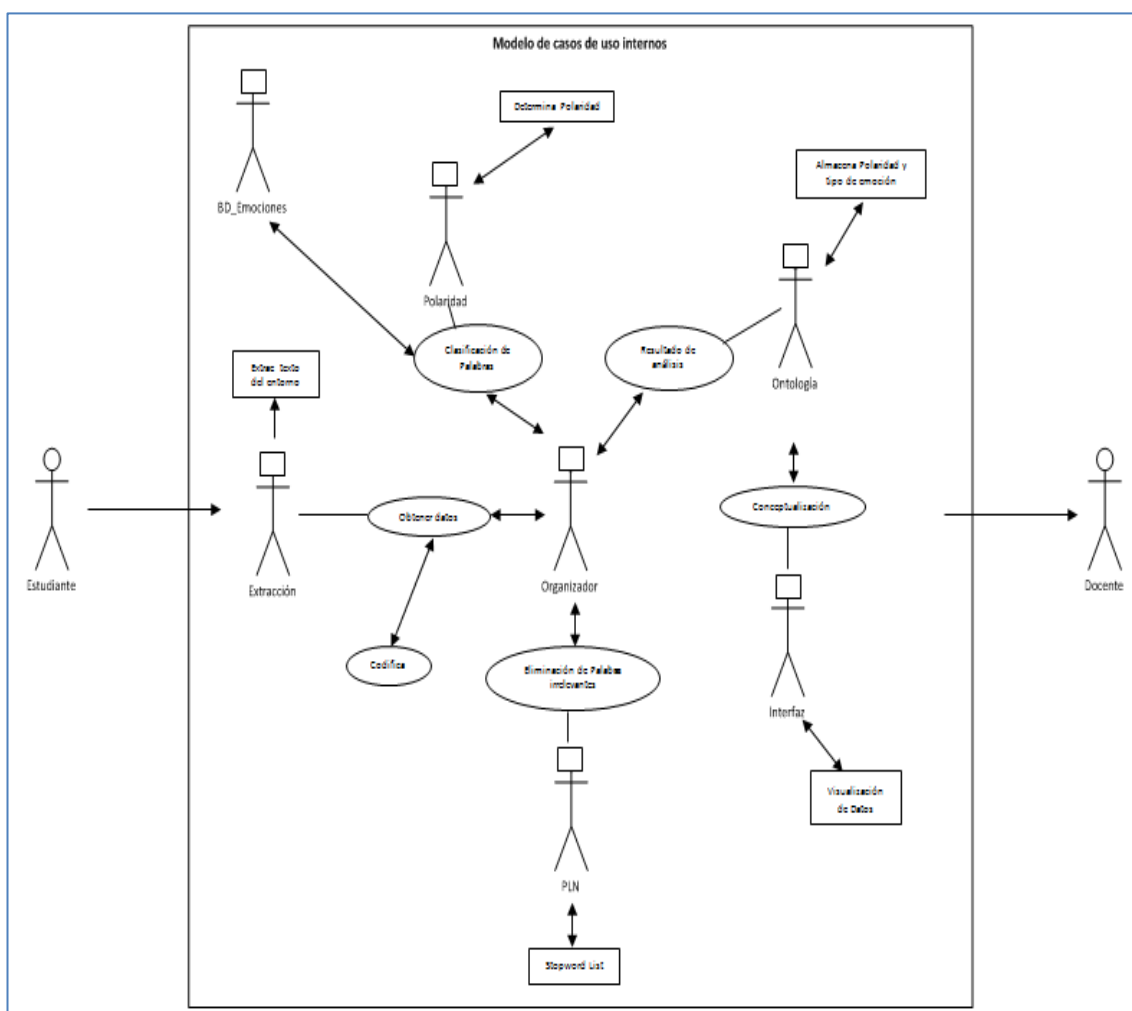


FIGURA 3.4 Casos de uso internos

▪ **Descripción de las Conversaciones**

Desde el punto de vista externo:

Conversación Obtener datos

Tipo

Obtención de información (texto)

Objetivo

Obtener texto (mensajes, foros, comentarios) del Entorno Virtual de Aprendizaje (EVA)

Agentes

Extracción

Iniciador

Estudiante

Servicio

Extracción de información del entorno

Descripción

El agente Extracción tiene como objetivo extraer información que ingresa el estudiante (texto), una vez obtenidos los datos los almacena en la BD_Emociones, para comunicarse con el agente Organizador para su posterior procesamiento.

Precondición

Ingreso de información (texto)

Postcondición

Datos extraídos

Condición-terminación

Información obtenida, o mensaje de error

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Codifica

Tipo

Obtención y procesamiento de información

Objetivo

Transformar el texto ingresado en formato legible

Agentes

Organizador

Iniciador

Agente Extracción

Servicio

Texto en formato legible

Descripción

El agente Organizador se encarga de transformar la información recolectada por el agente Extracción a un formato legible para enviarla al agente Segmentación.

Precondición

Solicitud de envío de información (texto)

Postcondición

Datos codificados en un formato específico.

Condición-terminación

Texto transformado a un formato legible

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Eliminación de palabras irrelevantes

Tipo

Procesamiento de información

Objetivo

Eliminar irrelevancias en la información recibida

Agentes

PLN

Iniciador

Agente Organizador

Servicio

Texto Limpio

Descripción

El agente PLN tiene como objetivo procesar el texto ingresado, mediante la utilización de un Stopword List.

Precondición

Solicitud de envío de información

Postcondición

Base de datos con texto procesado.

Condición-terminación

Texto sin irrelevancias

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Clasificación de Palabras

Tipo

Obtención de información

Objetivo

Determinar la polaridad del texto y envía esta información al docente

Agentes

Polaridad

Iniciador

Agente Organizador

Servicio

Polaridad del texto y visualización

Descripción

El agente Polaridad tiene como objetivo determinar si el texto ingresado es positivo o negativo y el tipo de emoción que presenta, mediante la utilización de un diccionario afectivo, raíces, intensificadores, verbos, negaciones, lo almacena en una BD_Emociones y en el Agente Ontología

Precondición

Recibir la información que el Agente Organizador recolecta del agente PLN

Postcondición

Polaridad del texto

Condición-terminación

Información de la polaridad del texto

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Envía entidades y conceptos

Tipo

Obtención de información

Objetivo

Conceptualizar y organizar en tripletas los datos obtenidos.

Agentes

Ontología

Iniciador

Agente Organizador.

Servicio

Conceptualización de texto

Precondición

Texto Polarizado

Postcondición

Base de datos de tripletas

Condición-terminación

Texto conceptualizado

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Presentación de resultados**Tipo**

Obtención de información

Objetivo

Presentar gráficamente los resultados obtenidos del análisis de texto.

Agentes

Interfaz

Iniciador

Agente Ontología

Servicio

Presentación de resultados

Precondición

Información almacenada en el Agente Ontología

Postcondición

Visualización gráfica de resultados

Condición-terminación

Visualización de resultados

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

▪ Descripción de los Protocolos de Comunicación

Una vez identificadas las conversaciones, se realiza una descripción detalladamente la descomposición de las intervenciones realizadas.

Conversación Obtener datos

Compuesta de

1. Estudiante ingresa información (Mensajes, foros, comentarios), al EVA la misma que es almacenada en BD_Emociones con entrada Id, texto.
2. Luego esta información es solicitada por el Agente Extracción.

Conversación Codifica

Compuesta de

1. El agente Extracción, obtiene la información de la BD_Emociones para codificar el texto en un formato legible.
2. Una vez codificado es enviado al Agente PLN para su procesamiento.

Conversación Eliminación de palabras irrelevantes

Compuesta de

1. El agente PLN recibe los datos codificados para procesarlos mediante el uso un Stopword List realiza la eliminación de las palabras irrelevantes
2. Luego de realizada la eliminación de palabras irrelevantes del texto, se envía los resultados a un archivo txt y a una base de datos BD_Emociones con entrada Id, texto, textoLimpio.
3. Una vez almacenado la información resultante el agente Organizador tiene acceso a la misma.

Conversación Clasificación de Palabras

Compuesta de

1. El agente Organizador envía la información obtenida del Agente PNL al Agente Polaridad, el mismo que determina si las palabras son positivas o negativas, mediante el uso de un diccionario afectivo.
2. Una vez obtenida la polaridad de las palabras estas se almacenan en un archivo txt, en el agente Ontología y en una BD_Emociones con entrada Id, palabra, polaridad.

Conversación Resultado de análisis

Compuesta de

1. El agente Organizador recolecta la información resultante del agente Polaridad, la misma que es enviada al agente Ontología.
2. El agente Ontología almacena la información resultante y la envía al agente Interfaz.

Conversación Conceptualización

Compuesta de

1. El agente Ontología recibe el texto polarizado y lo almacena, luego lo envía al agente Interfaz.
2. El agente Interfaz toma el texto polarizado y lo presenta al docente mediante una interfaz gráfica de usuario.

▪ **Descripción de las Intervenciones**

La descripción de intervenciones tiene como objetivo determinar el intercambio de mensajes entre los agentes. Para realizar la descripción de las intervenciones gráficamente utilizamos el diagrama de secuencia de mensajes.

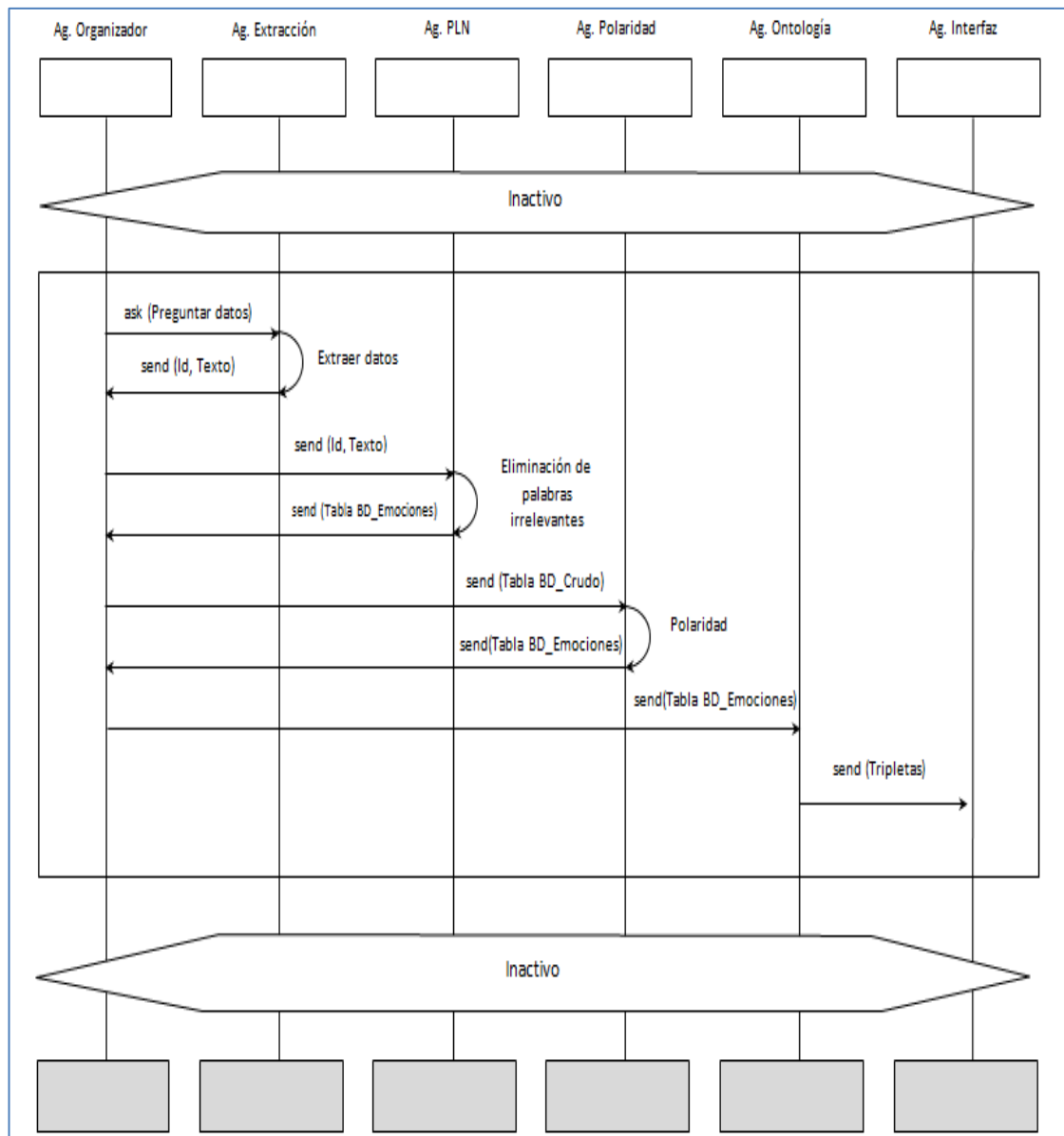


FIGURA 3.5 Diagrama de secuencias para las interacciones de la conversación

▪ **Canales básicos de Comunicación**

Una vez definidas todas las conversaciones, podemos determinar que mensajes son válidos entre los agentes, luego los representamos mediante un diagrama resumen como se muestra en la FIGURA 3.6.

- **Identificación de las Relaciones de Herencia**

El objetivo es modelar las relaciones “estáticas” entre los agentes que componen el sistema.

El agente Organizador es la clase principal del cual el resto de agentes hereda atributos principales para su funcionamiento.

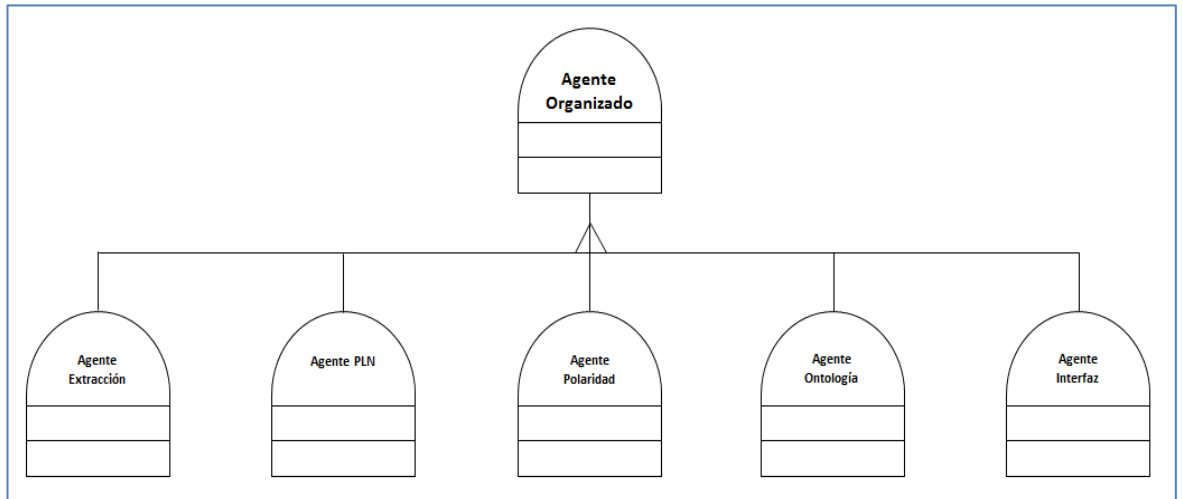


FIGURA 3.8 Jerarquía de Clases

- **Identificación de los Objetos del Entorno**

No hay un entorno muy definido dado que los agentes no son físicos, y no fue necesario definir sensores ni actuadores en las plantillas textuales de los agentes (Iglesias, 1998).

2.2.3 Estructura de Agentes de extracción de emociones en Video

Consiste en realizar un análisis de los componentes que formaran parte de la arquitectura, a partir de estos se puede realizar el diseño tanto físico y lógico, para la extracción de emociones en video.

- ✓ **Modelo de Agente**

Especifica las características de un agente, se orienta esencialmente a los servicios.

- **Identificación y descripción de los Agentes**

Para poder determinar los agentes del sistema se consideró la diferencia entre las tareas y los roles de los actores.

- o **Agentes Humanos**, son los actores activos identificados en la fase de conceptualización, y son: Docente y Estudiante
- o **Agentes Software**, una vez analizadas las tareas, se determinó los siguientes agentes: Extracción, Video, Ontología, Interfaz.

▪ **Identificación de las características de los agentes**

Tabla 3.7 Identificación del Agente Extracción de Video

Agente	Extracción	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Conocimiento de las entradas de video y su constante actualización.
Descripción	Agente software encargado de recolectar y almacenar todos los videos obtenidos de la interacción del estudiante con el docente a través del componente académico, se adapta a los cambios del entorno (constante actualización). Envía todos los datos recolectados a una base de datos (BD_Emociones), para su posterior procesamiento.	

Tabla 3.9 Identificación del Agente Video

Agente	Video	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Extrae información de la base de datos (BD_Emociones) para realizar el análisis del video.

Descripción	Este agente toma los datos que se almacena en la base (BD_Emociones) para realizar el análisis del video mediante la utilización EmotionDetection para extraer las emociones de las imágenes, cuyos resultados se almacenan en la base de datos BD_Emociones
--------------------	--

Tabla 3.10 Identificación del Agente Ontología de Video

Agente	Ontología	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Extrae los resultados de la BD_Emociones y almacena los resultados en una ontología.
Descripción	El agente Ontología es la entidad encargada de almacenar los resultados obtenidos del agente Video en una ontología, la cual nos almacena en una escala cuantitativa el tipo de emoción: de 5 (emoción positiva) a -5 (emoción negativa), y la emoción del rostro.	

Tabla 3.11 Identificación del Agente Interfaz de Video

Agente	Interfaz	
Tipo	Agente Software Inteligente	
Capacidades-razonamiento	Experiencia	Presenta la polaridad obtenida del video.

Descripción	Este agente tiene como entrada los resultados del análisis del video obtenido por el agente Video, según la polaridad que se obtenga, el agente envía un “alerta” al docente para que este pueda brindar retroalimentación al estudiante, mediante una interfaz gráfica.
--------------------	--

▪ **Distribución tareas-agentes**

La distribución de tareas nos permite comprobar los criterios de calidad. Para visualizar de mejor manera cuáles son las tareas que deben realizar cada uno de los agentes que conforman el sistema, es conveniente presentar la Tabla 3.12 en la que tenemos a cada uno de los agentes que conforman el sistema y las tareas que deben realizar.

Tabla 3.12 Distribución tareas-agentes de Video

	Obtener datos	Codificación de datos	Análisis de video	Conceptualización	Visualización de datos
Extracción	X	X			
Video			X	X	
Ontología				X	
Interfaz					X
BD_Emociones		X	X	X	

✓ **Identificación y descripción de los Objetivos**

Para identificar los objetivos partimos de la asignación de tareas a cada agente, y tenemos:

Objetivo: Adaptarse al usuario y codificar datos

Parámetros-entrada

Videos

Parámetros-salida

Video en un formato legible y lo almacena

Condición-activación

Inicio de la transmisión de un video

Condición- finalización

Cuando no existen nuevas entradas de video

Descripción

El agente Extracción tiene como objetivo recolectar y almacenar todos los videos obtenidos de la interacción del estudiante con el docente, extrae los videos y los almacena en BD_Emociones en un formato legible.

Objetivo: Análisis de video y conceptualización

Parámetros-entrada

Toma información de la base de datos obtenida del entorno (videos) por el agente de Extracción

Parámetros-salida

Polaridad del video y emoción detectada en el rostro

Condición-activación

Extrae la información obtenida por el agente Extracción para procesarla

Condición- finalización

Que no exista información pendiente de procesar

Descripción

Este agente esta encargado de extraer los datos que se almacena en la base (BD_Emociones), realiza el análisis de imágenes mediante EmotionDetection y envía la polaridad y la emoción detectada en el rostro del estudiante, que se almacena en el Agente Ontología.

Objetivo: Visualización de datos

Parámetros-entrada

Toma información de la del Agente Ontología (Polaridad)

Parámetros-salida

Polaridad del video y emoción detectada en el rostro mediante una interfaz gráfica

Condición-activación

Extrae la información obtenida por el agente Ontología para presentarla visualmente

Condición- finalización

Que no exista información pendiente de presentar

Descripción

Este agente esta encargado de presentar mediante una interfaz gráfica los resultados de la polaridad del video almacenados en el Agente Ontología.

✓ **Modelo de Tareas**

Se describe la descomposición funcional del sistema, las actividades necesarias que debe realizar para cumplir con sus objetivos.

Tarea Análisis de Sentimientos

Objetivo

Obtener estados emocionales del estudiante

Descripción

En esta tarea de determina el estado emocional de los estudiantes en un curso virtual

Entrada

Video

Salida

Polaridad del video

Precondición

Video recibido

Supertarea

Procesamiento de video 1

Subtareas

Obtener datos 1.1.1, Codificar datos 1.1.2, Análisis de video 1.1.3, Conceptualización 1.1.4, Visualización de datos 1.1.5.

Tipo de descomposición

Temporal/Funcional

Ingrediente PolaridadEmociones

Descripción

Polaridad de emociones implica analizar las emociones presentes en video, para obtener su polaridad y determinar si el estudiante necesita retroalimentación en algunos temas del componente académico.

- **Descomposición de tareas**

Mediante diagramas de Definición Integrada para Modelado de Funciones (IDEF):



FIGURA 3.9 Descomposición de la Tarea T1: Análisis de Sentimientos Video

- ✓ **Modelo de Coordinación**

- **Identificación de las Conversaciones**

Para ilustrar las conversaciones utilizaremos la técnica de casos de uso internos.

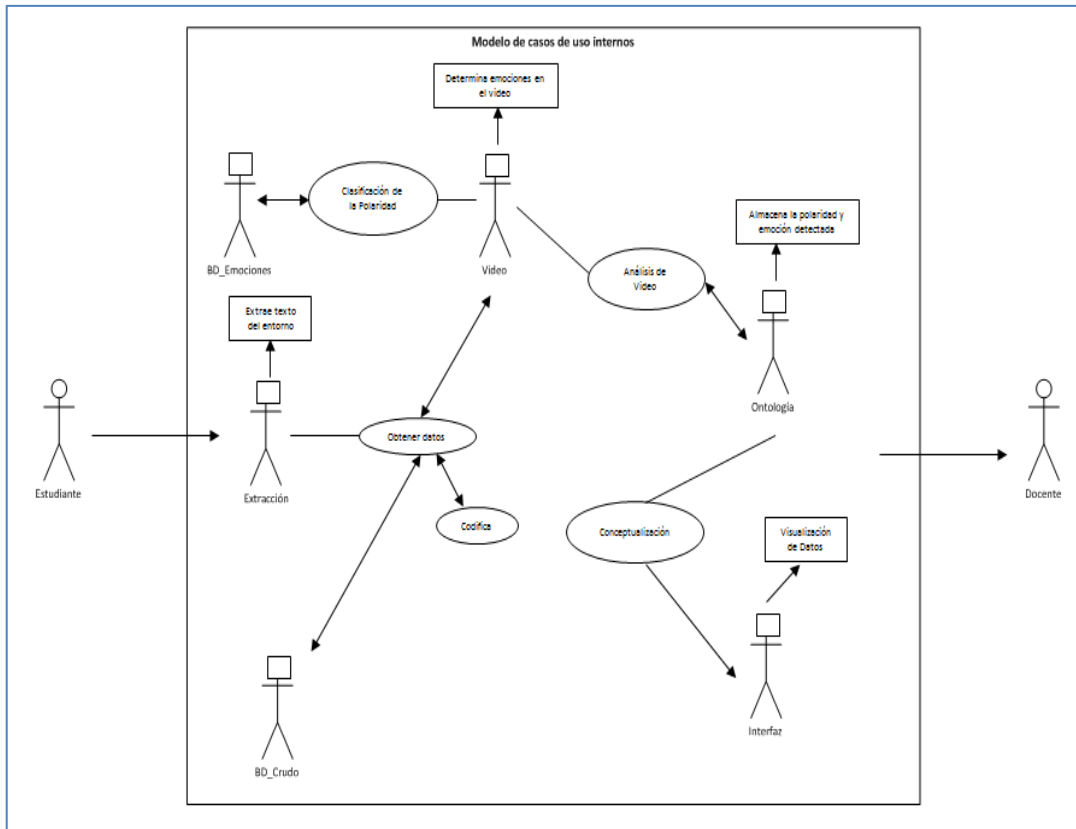


FIGURA 3.10 Casos de uso internos Análisis de Video

▪ Descripción de las Conversaciones

Desde el punto de vista externo:

Conversación Obtener datos

Tipo

Obtención de información (video)

Objetivo

Obtener los videos del entorno virtual de aprendizaje

Agentes

Extracción

Iniciador

Estudiante

Servicio

Extracción de información del entorno

Descripción

El agente Extracción tiene como objetivo extraer información que ingresa el estudiante (video), una vez obtenidos los datos los almacena en la

BD_Emociones, para comunicarse con el agente Video para su posterior procesamiento.

Precondición

Ingreso de información (video)

Postcondición

Datos extraídos

Condición-terminación

Información obtenida, o mensaje de error

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Codificar

Tipo

Obtención y procesamiento de información

Objetivo

Transformar el video ingresado en formato legible

Agentes

Extracción

Iniciador

Agente Extracción

Servicio

Video en formato legible

Descripción

El agente Extracción también se encarga de transformar la información recolectada a un formato legible para enviarla al agente Video.

Precondición

Solicitud de envío de información (video)

Postcondición

Datos codificados en un formato específico.

Condición-terminación

Video transformado a un formato legible

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Clasificación de Polaridad

Tipo

Obtención de información

Objetivo

Envía la información de la polaridad obtenida a la base de datos (BD_Emociones)

Agentes

Video

Iniciador

Video

Servicio

Polaridad del video

Descripción

El agente Video tiene como objetivo determinar si el video ingresado es positivo o negativo, lo almacena en una BD_Emociones y envía los resultados al agente Ontología para que almacene la emoción detectada y su polaridad.

Precondición

Recibir la información resultante del procesamiento del Agente Video

Postcondición

Polaridad del video

Condición-terminación

Información de la polaridad del video

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Análisis de Video

Tipo

Obtención de información

Objetivo

Envía la polaridad del video al Agente Ontología

Agentes

Video

Iniciador

Video

Servicio

Polaridad del video y emociones encontradas

Descripción

El agente Video tiene como objetivo determinar si el texto ingresado es positivo o negativo, el resultado enviado al agente Ontología es la polaridad en forma cuantitativa así como la emoción detectada en la imagen.

Precondición

Recibir la información resultante del Agente Video

Postcondición

Polaridad del video

Condición-terminación

Información de la polaridad del video (emoción y polaridad)

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

Conversación Conceptualización (Envía entidades y conceptos)

Tipo

Obtención de información

Objetivo

Conceptualizar y organizar en tripletas los datos obtenidos.

Agentes

Ontología

Iniciador

Agente Video

Servicio

Conceptualización del resultado del análisis del video

Descripción

El agente Ontología recibe el resultado del procesamiento del video realizado por el agente Video y almacena la emoción obtenida del análisis de imágenes y su polaridad para enviarlo al Agente Interfaz para que sea presentado al usuario (docente).

Precondición

Video analizado

Postcondición

Base de datos de tripletas

Condición-terminación

Conceptualizado del resultado

Tiempo-ejecución

Depende del tamaño de los datos con el que se alimenta el proceso

▪ **Descripción de los Protocolos de Comunicación**

Cuando se identifican las conversaciones se describe detalladamente la descomposición de las intervenciones realizadas.

Conversación Obtener datos

Compuesta de

1. El estudiante ingresa información (Videos) la misma que es almacenada en BD_Emociones con entrada Id, video.
2. Luego esta información es solicitada por el Agente Extracción.

Conversación Codificar

Compuesta de

1. El agente Extracción, obtiene la información de la BD_Emociones para codificar el video en un formato legible.
2. Una vez codificado es enviado al Agente Video para su procesamiento.

Conversación Clasificación de Polaridad y Análisis de Video

Compuesta de

1. El agente Video recibe los datos codificados para procesarlos mediante el uso de EmotionDetection para obtener la polaridad de las emociones encontradas y el tipo de emoción presente en el video.
2. Luego de realizado este procesamiento, se envía los resultados a BD_Emociones con entrada Id, polaridad, emoción.
3. También estos resultados son enviados al agente Ontología para su almacenamiento.

Conversación Conceptualización

Compuesta de

3. El agente Ontología recibe los resultados obtenidos por el agente Video.
4. El agente Ontología toma la polaridad obtenida y la emoción resultante y crea conceptos y a partir de estos puede enviar al agente Interfaz los resultados que debe presentar.

▪ Descripción de las Intervenciones

La descripción de intervenciones tiene como objetivo determinar el intercambio de mensajes entre los agentes. Para realizar la descripción de las intervenciones gráficamente utilizamos el diagrama de secuencia de mensajes.

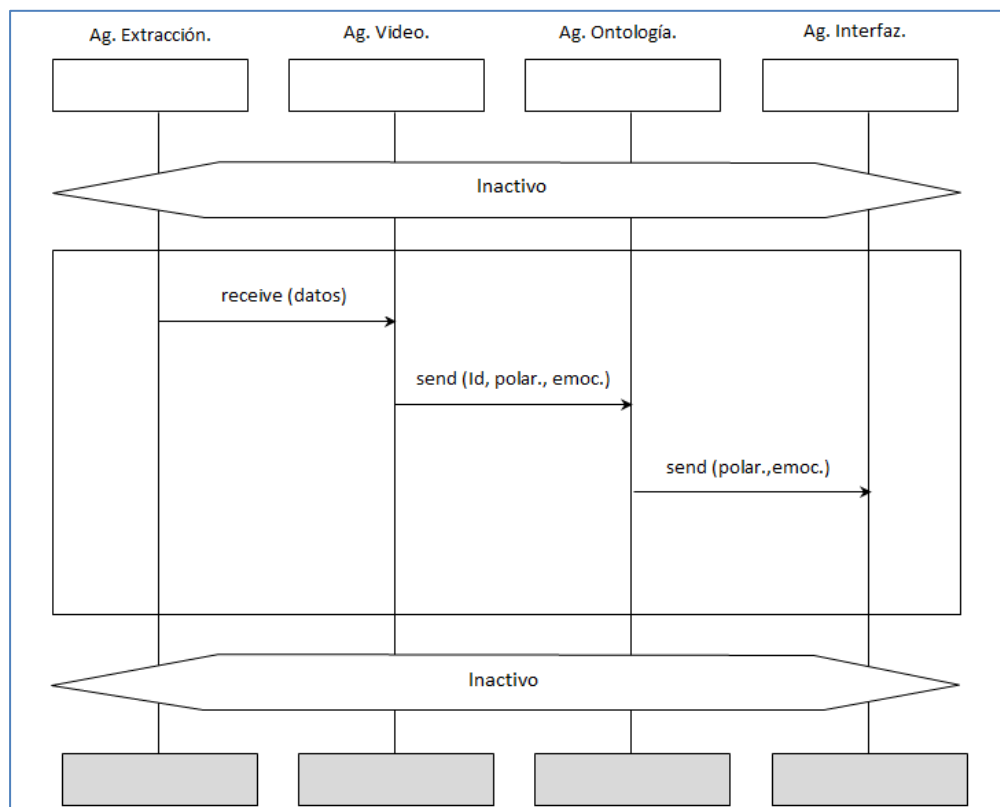


FIGURA 3.1111 Diagrama de secuencias para las interacciones de la conversación de Video

▪ Canales básicos de Comunicación

Una vez definidas todas las conversaciones, podemos determinar que mensajes son válidos entre los agentes, luego los representamos mediante un diagrama resumen como se muestra en la FIGURA 3.12.

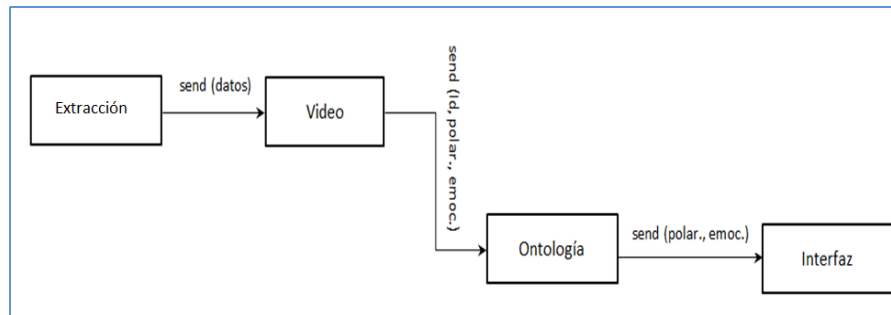


FIGURA 3.12 Canales básicos de comunicación y flujo de conversaciones entre agentes de Video

✓ **Modelo de Comunicación**

Modela las interacciones hombre-máquina, se descompone el diálogo entre el programa y el usuario utilizando un diagrama de secuencia de mensajes. Se considera los factores humanos en las interacciones.

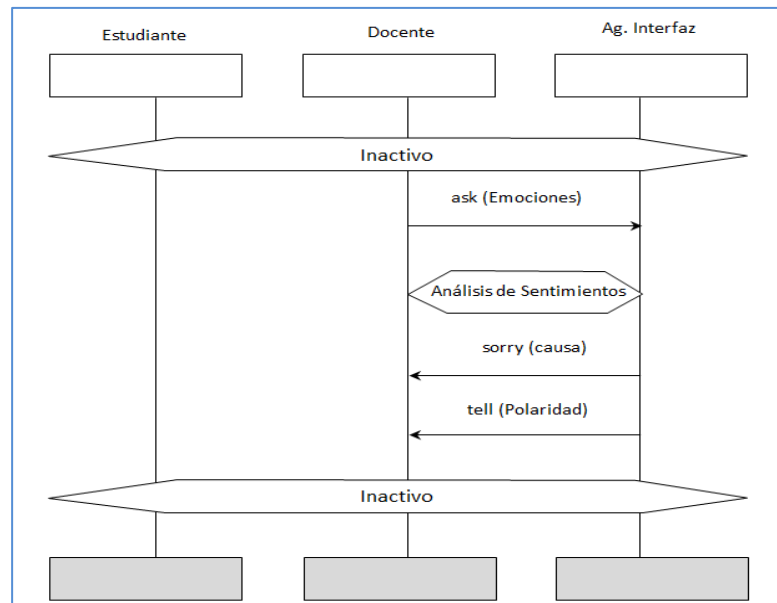


FIGURA 3.13 Interacciones genéricas de la conversación de Video

✓ **Modelo de la Organización**

El objetivo es modelar las relaciones “estáticas” entre los agentes del sistema.

▪ **Identificación de los Objetos del Entorno**

No hay un entorno muy definido dado que los agentes no son físicos, y no fue necesario definir sensores ni actuadores en las plantillas textuales de los agentes (Iglesias, 1998).

CAPÍTULO 4

Propuesta de la Arquitectura de Agentes

4.1 Introducción

En este capítulo se detalla el diseño de la arquitectura propuesta, cuya finalidad es describir los componentes que cumplen con los requerimientos del Modelo de Análisis (Capítulo 3), se muestra su funcionamiento lógico y su estructura.

Mediante el uso de esquemas de agentes simples, se pretende demostrar el funcionamiento de los agentes que intervienen en la Arquitectura, la forma en que percibe su entorno, procesa dichas percepciones y responde de forma correcta.

También se detalla los componentes que constituyen la arquitectura, se presenta el modelo de arquitectura propuesta, así como su comunicación entre agentes y finalmente se transforman las especificaciones de los modelos anteriores a un lenguaje de programación.

4.2 Modelo de Diseño

En esta fase se depura los modelos anteriores y se muestra la arquitectura seleccionada para los agentes y las funciones de sus componentes o módulos que deben implementarse en la arquitectura propuesta, se determina que arquitectura de agentes es apropiada para nuestro sistema y el lenguaje de implementación.

4.2.1 Esquema del Agente detección de Emociones en Texto

En la Figura 4.1 se muestra el esquema general del Agente Simple de detección de emoción en texto, en el mismo se puede observar como los agentes que componen la arquitectura reciben las percepciones de su entorno (EVA), mediante un analizador de patrones captan las señales (sensores) y son enviadas a los agentes se encargan de realizar el procesamiento de la información recibida y finalmente presentan el tipo de emoción encontrada.

La Arquitectura recibe como parámetro de entrada texto obtenido de la mensajería, red social y foros, luego los procesa mediante el uso de un stopword list, diccionario afectivo, intensificadores, verbos, negaciones.

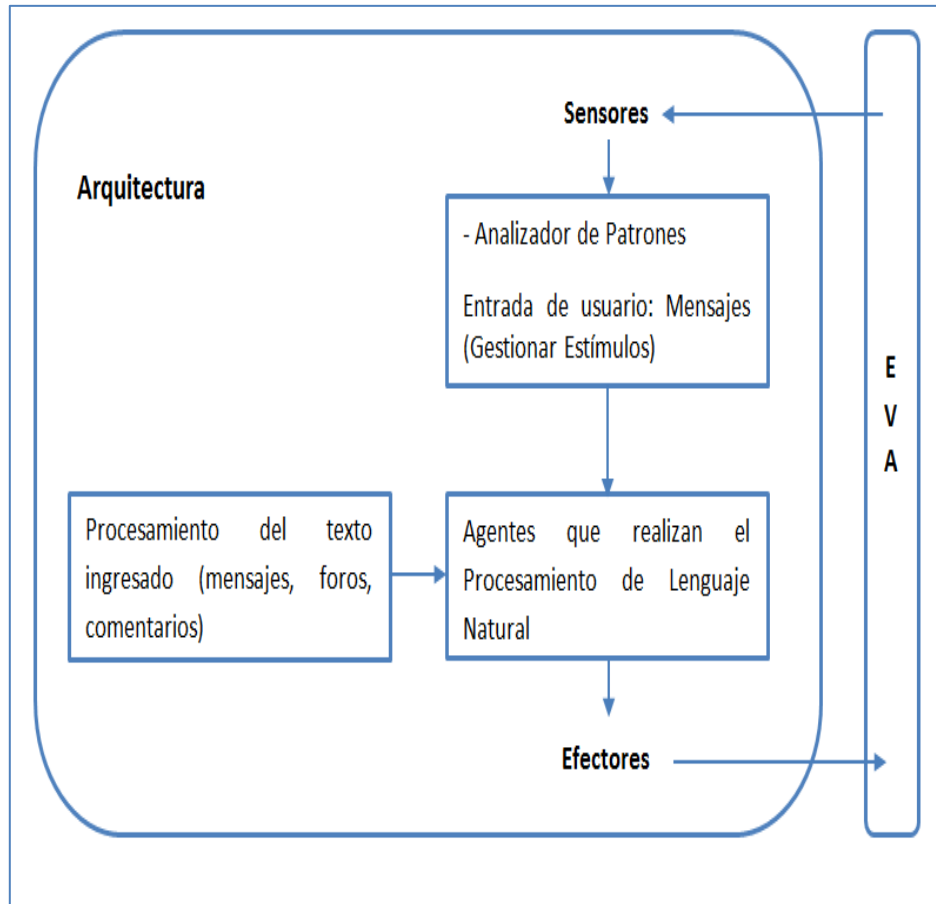


FIGURA 4.1 Esquema de agente simple detección de emociones en texto

4.2.2 Esquema del Agente detección de Emociones en Video

En la Figura 4.2 se presenta el esquema del agente detección de emociones en video. Trabaja de forma similar al Agente Simple de detección de emoción en texto, con la única diferencia utiliza una cámara web (sensores) para obtener las percepciones del entorno, las mismas que son enviadas a los agentes se encargan de realizar el procesamiento imágenes y finalmente presentan el resultado.

La Arquitectura recibe como parámetro de entrada videos, luego los procesa mediante EmotionDetection y finalmente nos presenta el tipo de emoción encontrada.

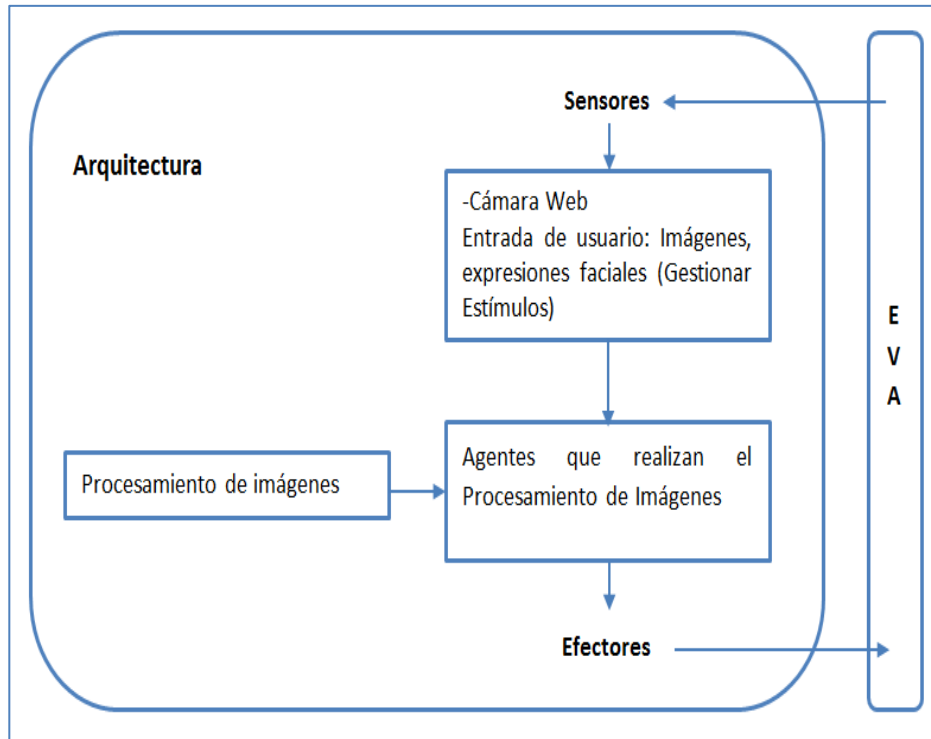


FIGURA 4.2 Esquema de Agente Simple detección de emoción en video

4.3 Componentes

Dentro de los componentes que constituyen la arquitectura multiagente tenemos:

✓ El agente emoción en texto y video presentan componentes en común:

- Estudiante
- Docente
- Almacenamiento de resultados

✓ Para el agente emoción en texto:

- Obtención y Almacenamiento de datos
- Procesamiento de Texto

✓ Para el agente emoción en video:

- Obtención y Almacenamiento de datos
- Procesamiento de Imágenes

4.4 Descripción de componentes

✓ Componentes comunes entre el agente emoción en texto y video:

- **Estudiante:** Es la persona que interactúa con el entorno virtual de aprendizaje (EVA), de su interacción se obtiene los mensajes, foros, comentarios, y mensajes de la red social, para su posterior procesamiento.
- **Docente:** Es la persona que interactúa con la aplicación, para realizar el análisis de las emociones de los estudiantes.
- **Almacenamiento de resultados:** Es el almacenamiento de los resultados del análisis de texto, en BD_Emociones y en la Ontología, lo que almacenan es el tipo de emoción y el valor cuantitativo.

✓ Componentes del agente emoción en texto:

- **Obtención y Almacenamiento de datos:** Mediante el Agente Extracción, se obtiene información del entorno (mensajes, foros, comentarios), y se almacena en la base de datos BD_Emociones, luego es enviada al agente Organizador quien se encarga de enviar todo lo extraído al resto de agentes y coordina su funcionamiento.
- **Procesamiento de Texto:** Aquí se describe los agentes que se implementaron para el procesamiento de texto. El agente que interviene en este componente es el de PLN y los resultados que proporcionan este agente son almacenados en la BD_Emociones, y el Agente Organizador se encarga de enviarlo secuencialmente al agente Polaridad para que pueda procesar la información y presente el tipo de emoción encontrada.

✓ **Componentes del agente emoción en video:**

- **Obtención y Almacenamiento de datos:** Mediante el Agente Extracción, se obtiene información del entorno (videos), y se almacena en la base de datos BD_Emociones, esta información es enviado al componente Procesamiento de Imágenes.

- **Procesamiento de Imágenes:** La herramienta que se plantea utilizar para el análisis de imágenes es **EmotionDetection** que permite hacer el análisis de las imágenes, cuyo resultado se almacena en BD_Emociones.

4.5 Arquitectura Propuesta

Basándonos en los requisitos obtenidos en los modelos de la metodología MAS_COMMONKADS se propone dos Arquitecturas basadas en el estándar FIPA: detección de emociones en Texto y detección de emociones en Video.

✓ Arquitectura detección de Emociones en Texto

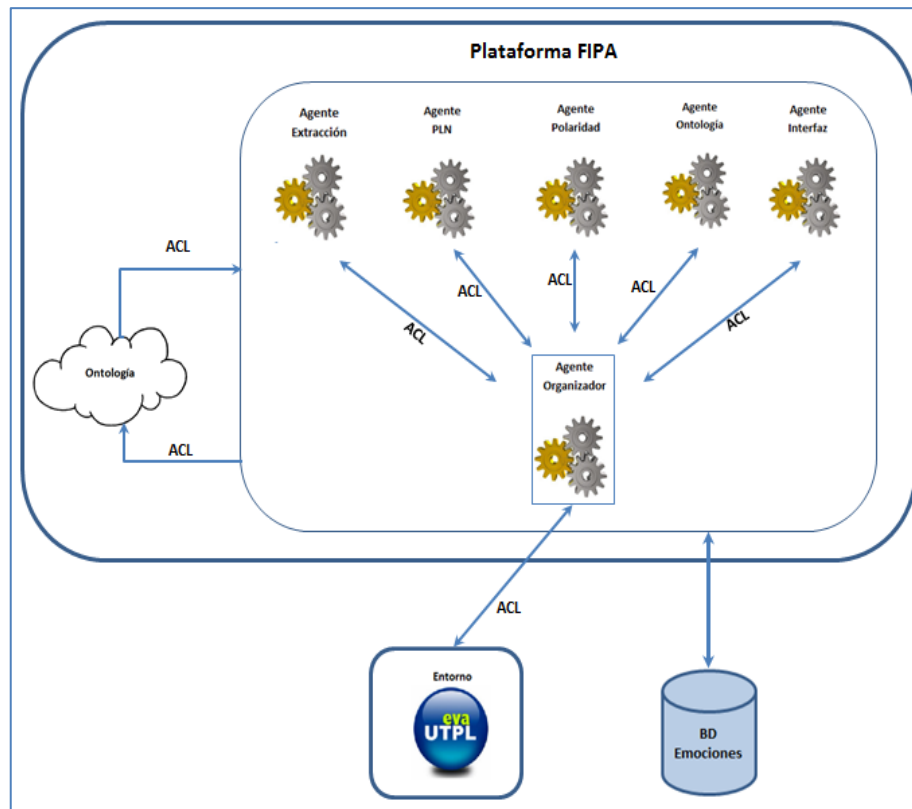


FIGURA 4.3Arquitectura propuesta detección de Emociones en Texto

✓ Arquitectura detección de Emociones en Video

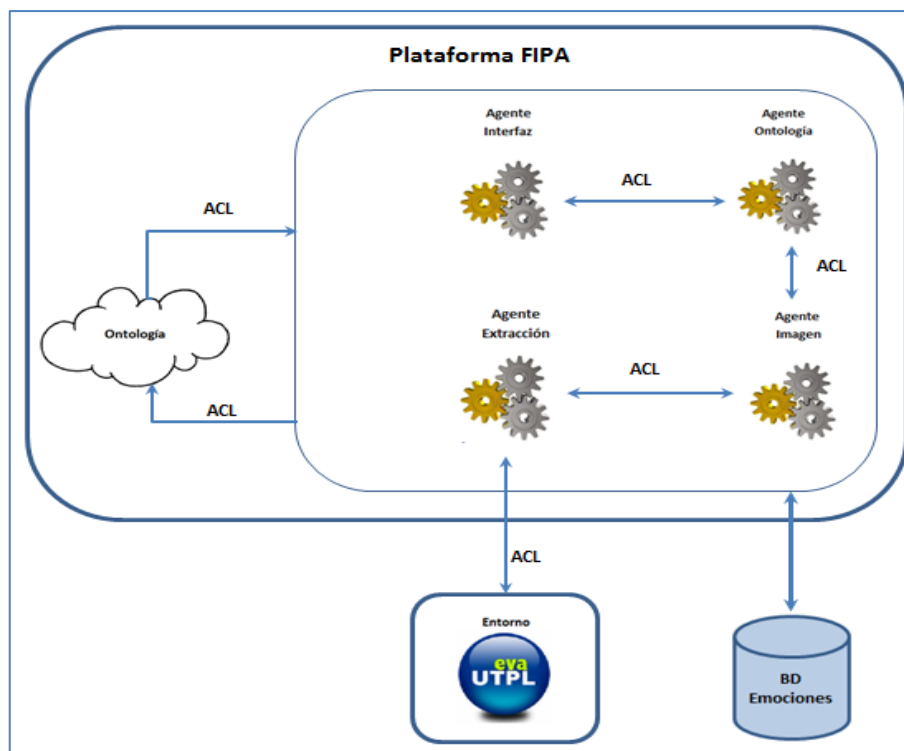


FIGURA 4.4 Arquitectura propuesta detección de Emociones en Video

4.5.1 Comunicación entre Agentes

El lenguaje de comunicación entre agentes es un tema importante, ya que nos permite compartir y enviar mensajes entre los agentes, pero es necesario que estos mensajes tengan significado (contenido semántico), por esta se considera la utilización de ACL-FIPA.

ACL-FIPA es un lenguaje estandarizado para la comunicación de agentes, su principal objetivo es darle sentido semántico a los mensajes intercambiados entre los agentes. Para conseguir esto se basaron principalmente en los actos comunicativos para solicitar una acción a un agente. Para realizar una acción el agente "A" envía un mensaje *request* al agente "B"; el agente que recibe el mensaje puede aceptar (*accept*) o rechazar (*refuse*) la acción. En caso de que el agente "B" acepte la petición tiene que notificar e indicar al agente "A" cuando la realización de la acción finalice a través del mensaje *agree*. En caso de que la realización de la acción falle se notificara al agente "A" mediante el envío de un mensaje *failure* (Pinzón, 2010).

4.6 Desarrollo de la Arquitectura de detección de emociones en Texto

4.6.1 Lenguaje de Programación

El lenguaje de programación seleccionado para la implementación de la arquitectura de detección de emociones en texto es JAVA, básicamente se utilizó del framework JADE desarrollado para la creación de sistemas multiagente.

4.6.2 Implementación de los Agentes

Con la finalidad de ilustrar el procesamiento de datos que realiza cada agente se utilizó la técnica de diagramas de flujo para la representación gráfica del proceso, además se presenta parte del pseudocódigo desarrollado para la implementación de los agentes

- **Diagramas de Agentes**

- ✓ **Agente Extracción**

Este agente está encargado de recolectar los datos (texto), presenta una condición: si existen los datos pasa a almacenarlos en la base de datos BD_Emociones y lo envía al siguiente agente para que procese los datos recibidos, caso contrario finaliza el proceso.

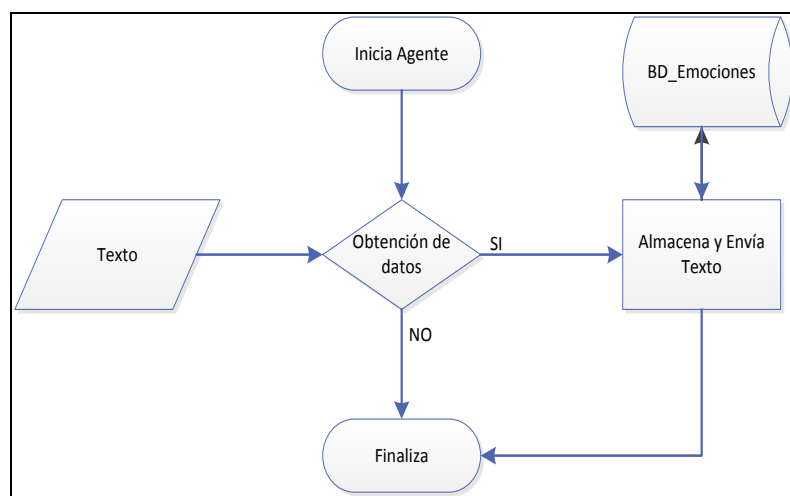


FIGURA 4.5 Diagrama de flujo Agente Extracción

✓ **Agente PLN:**

Toma los datos obtenidos por el agente Extracción y realiza la eliminación de palabras irrelevantes o sin significado, mediante la utilización de un StopWord List. El resultado final se almacena en la base de datos BD_Emociones y en un archivo txt.

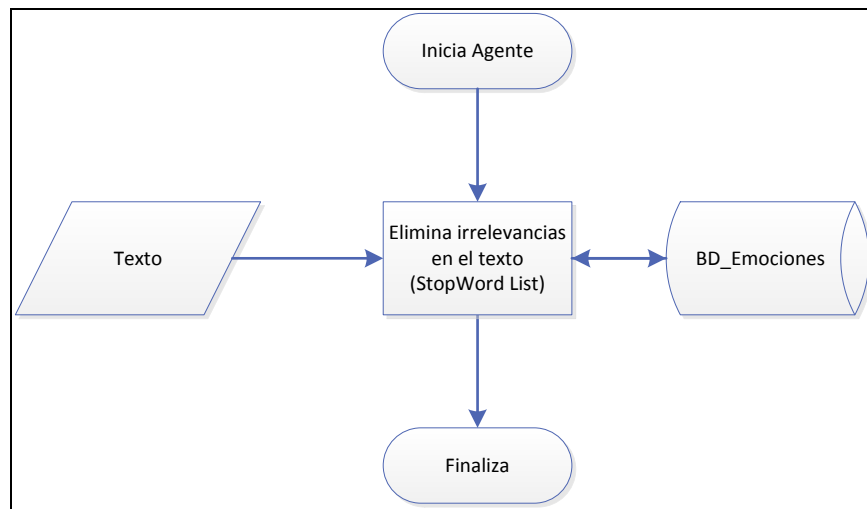


FIGURA 4.6 Diagrama de flujo Agente Segmentación

✓ **Agente Polaridad**

Determina la polaridad del texto y el tipo de emoción, toma como parámetro de entrada el resultado del agente PLN, utiliza intensificadores, negaciones, verbos, palabras raíces y un diccionario afectivo, para poder obtener la polaridad del texto.

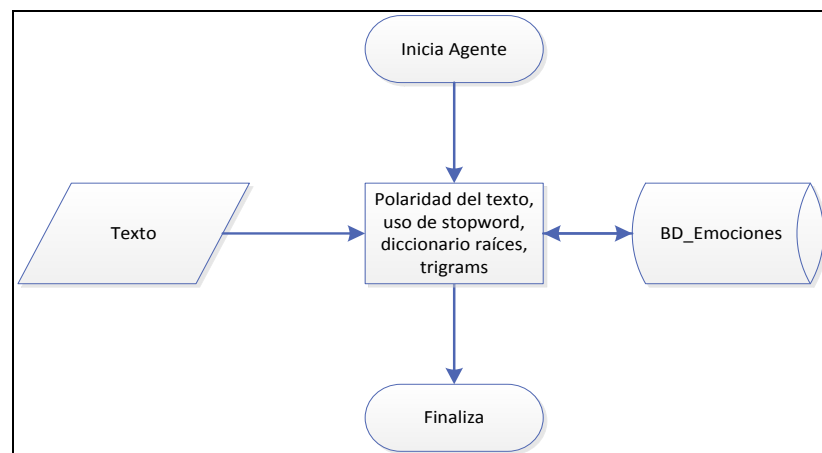


FIGURA 4.7 Diagrama de flujo Agente Polaridad

4.6.3 Desarrollo de la Arquitectura

✓ Herramienta gráfica

A continuación se presenta la herramienta gráfica RMA (Remote Monitoring Agent) propia de JADE, en la cual podemos inicializar los agentes. Inicializamos los agentes que intervienen en la arquitectura, Ontología, Polaridad e Interfaz.

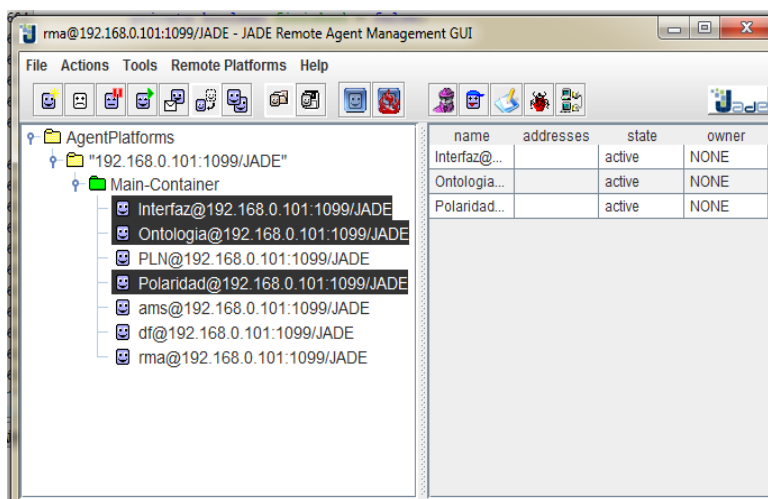


FIGURA4.8 Interfaz gráfica del RMA

El sistema multiagente nos permite realizar el análisis de texto, obtiene su polaridad y el tipo de emoción detectada, procesa varios mensajes a la vez y se presenta el resultado por cada mensaje analizado y lo almacena en archivo .txt.

En el botón Cargar, seleccionamos el archivo que deseamos procesar, luego damos clic en el botón Realizar- Proceso, y podemos visualizar en la interfaz gráfica el resultado y también en un archivo .txt. El tiempo de ejecución y muestra de resultados dependerá del tamaño del corpus con el que trabajemos. Con fines educativos se tomó varios mensajes de la Red Social del Eva y se obtuvo los siguientes resultados:

Resultados obtenidos con 30 mensajes:

- Los mensajes utilizados para las pruebas se encuentra en los Anexos1-A

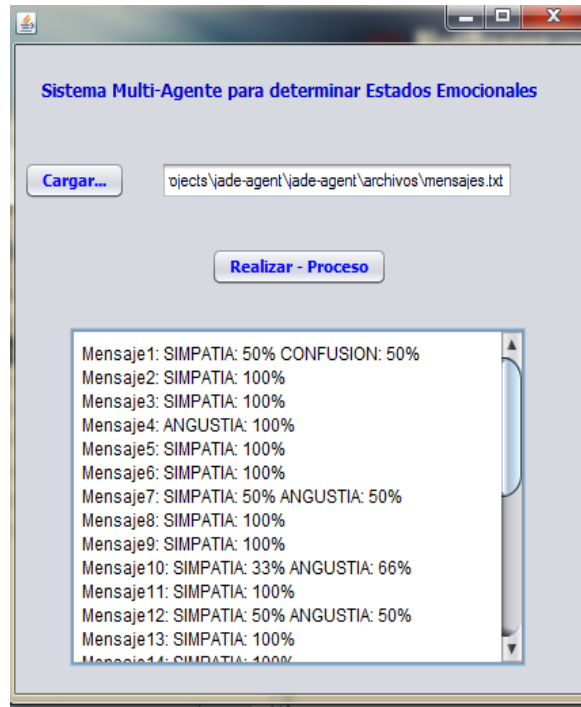


FIGURA 4.9 Resultados del Sistema

Como podemos observar en la FIGURA 4.9 el resultado obtenido se muestra por cada mensaje analizado, se realizó varias pruebas y se tiene como resultado global que las emociones simpatía, confusión, angustia y frustración, están presentes o influyen con mayor frecuencia en el aprendizaje de los estudiantes, lo cual trae como consecuencia el bajo rendimiento de los mismos, por esta razón es conveniente que el docente conozca esta información para que puede brindar una retroalimentación a los estudiantes en los componentes académicos.

Además, cabe recalcar que el resultado obtenido del procesamiento de texto por parte de nuestro sistema multiagente fue comparado con un análisis previo que se realizó mediante la utilización de escenarios, de los cuales se obtuvo varias características que nos ayudan a determinar la polaridad de las emociones y el tipo de emoción, dichas características fueron plasmados en el sistema multiagente.

CONCLUSIONES

Mediante el desarrollo e implementación de la arquitectura multiagente, se puede concluir lo siguiente:

- ✓ El Análisis de Sentimientos intenta clasificar los documentos según su polaridad, mediante la utilización de PLN, para determinar ciertos patrones que nos permitan conocer que emociones afectan o están presentes en el aprendizaje.
- ✓ La metodología MAS_CommonKADS nos permite el desarrollo de sistemas multiagente mediante la utilización de modelos, para obtener todos los requerimientos necesarios para la implementación de la Arquitectura, cabe recalcar que esta metodología nos ofrece un marco de especificaciones independiente de la implementación.
- ✓ JADE es un marco de desarrollo para la creación de sistemas multiagente, nos brinda interoperabilidad, portabilidad y está basado en los estándares FIPA para definir protocolos de comunicación, además nos permite trabajar con ontologías y permite el trabajo peer-to-peer.
- ✓ Es conveniente la utilización de un StopWord, para la eliminación de palabras irrelevantes que contienen los mensajes, ya que no representan ningún tipo de emoción.
- ✓ El planteamiento de escenarios es un punto importante para el desarrollo de este proyecto, porque nos ayuda a conocer e interpretar las alteraciones lingüísticas que presentan los mensajes, de esta forma se pueden obtener ciertas palabras clave que nos ayudan a determinar los tipos de emociones presentes en los mismos.
- ✓ La aplicación de la metodología MAS_CommonKADS nos permite obtener todos los requerimientos necesarios para implementar la Arquitectura que se plantea.
- ✓ Las emociones detectadas mediante el análisis de los mensajes con mayor ocurrencia son: Confusión y Angustia. Estas emociones nos indican que los estudiantes necesitan una retroalimentación en algunos temas de estudio.

RECOMENDACIONES Y TRABAJOS FUTUROS

- ✓ Es necesario seguir una metodología de desarrollo de sistemas multiagente, ya que en su desarrollo se establece todos los requerimientos necesarios para su posterior implementación.
- ✓ Para comprender mejor en análisis textual, es conveniente conocer algunos temas concernientes a ciertas alteraciones de las formas lingüísticas, que se encuentran presentes en los mensajes.
- ✓ Plantear escenarios es importante para poder obtener ciertas características que influyen en los estados emocionales de los estudiantes.
- ✓ Para la implementación de la arquitectura propuesta, es necesario analizar todas las herramientas disponibles y elegir la que mejor se adapte a nuestros requerimientos.
- ✓ Al ser un tema de actualidad tiene mucho potencial, se puede desarrollar aplicaciones móviles para realizar análisis de sentimientos, y según el estado de ánimo del usuario, cambiar el tono de llamada o mensaje de su teléfono móvil.
- ✓ Se puede desarrollar una aplicación educativa, mediante el uso de las redes sociales como punto de partida para el análisis psicológico y emocional de los estudiantes.
- ✓ Además, realizar un análisis del rendimiento educativo, según el estado emocional del estudiante.

BIBLIOGRAFÍA

- Akbiyik, C. (2010). ¿Puede la informática afectiva llevar a un uso más efectivo de las Tecnologías de la Información y de la Comunicación (TIC) en la Educación? *Revista de Educación*(352), 190.
- Baggia, P., Burkhardt, F., Pelachaud, C., Peter, C., & Zovato, E. (2012). *Emotion Markup Language (EmotionML) 1.0*. Obtenido de W3C incubator group report, World Wide Web Consortium: <http://www.w3.org/TR/emotionml/>
- Barrett, L. F., & Russell, J. A. (1999). *The Structure of Current Affect: Controversies and Emerging Consensus*. Obtenido de American Psychological Society: <http://www.affective-science.org/pubs/1999/FBRussell99.pdf>
- Bellifemine, F., Caire, G., & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Wiley.
- Blanco, A., & Carracedo, A. (2009). *Inteligencia Emocional*. Obtenido de Universia: <http://dspace.universia.net/bitstream/2024/195/1/Trabajo+imprimible.pdf>
- Borges, F. (mayo de 2005). *La frustración del estudiante en línea. Causas y acciones preventivas*. Obtenido de DIGITHUM: <http://www.uoc.edu/digithum/7/dt/esp/borges.pdf>
- Bratman, M. (1987). *Intentions, Plans, and Practical Reason*. Harvard University Press.
- Breiman, L. (1994). Obtenido de <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=506D5DF160BDEDFE92B85A1261E1549F?doi=10.1.1.32.9399&rep=rep1&type=pdf>
- Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Boca Raton, Florida: CHAPMAN & HALL.
- Cabello, E., Conde, C., Martín de Diego, I., & Serrano, Á. (2006). Técnicas de Reconocimiento Automático de Emociones. *Revista Electrónica Teoría de la Educación. Educación y Cultura en la Sociedad de la Información. Universidad de Salamanca*, 107-127.
- Carrillo, J. (2011). *Un Modelo Lingüístico-Semántico Basado en Emociones para la Clasificación de Textos según su Polaridad e Intensidad*. Obtenido de Natural Language Processing and Information Retrieval Group at UNED: http://nlp.uned.es/~jcalbornoz/papers/PhD_Thesis_2011.pdf

- Chuang, Z.-J., & Wu, C.-H. (2004). *Multi-Modal Emotion Recognition from Speech and Text*. Obtenido de The Association for Computational Linguistics: <http://www.aclweb.org/anthology-new/O/O04/O04-3004.pdf>
- Corchado, J. M. (2005). *Universidad de Salamanca: BISITE (Bioinformática, Sistemas Inteligentes, Tecnología Educativa)*. Obtenido de Modelos y Arquitecturas de Agentes: [http://bisite.usal.es/archivos/c1%20\(1\).pdf](http://bisite.usal.es/archivos/c1%20(1).pdf)
- Cortijo, F. (2000). *Técnicas supervisadas II: Aproximación no paramétrica*. Obtenido de Instituto de Ingeniería Eléctrica. Universidad de Granada.: http://iie.fing.edu.uy/ense/asign/recpat/material/tema3_00-01/node25.html
- DeLoach, S. (2011). *Analysis and Design using MaSE and agentTool*. Obtenido de CiteSeerX: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=EE1D62BF60F8D3FCD31655B74C51F55D?doi=10.1.1.155.3813&rep=rep1&type=pdf>
- Fakhraie, N. (2012). *What's in a Note? Sentiment Analysis in Online Educational Forums*. Obtenido de https://tspace.library.utoronto.ca/bitstream/1807/30086/1/Fakhraie_Seyedeh_Najmeh_201111_MA_thesis.pdf
- Fausett, L. (1994). *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications*. New Jersey, USA: Prentice-Hall.
- Ferber, J. (1999). *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*. Harlow: Addison Wesley Longman.
- Freund, Y., & Schapire, R. (1996). *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. Obtenido de Journal of Computer and System Sciences: http://www.face-rec.org/algorithms/Boosting-Ensemble/decision-theoretic_generalization.pdf
- Gevers, T. (2008). *Visual Recognition*.
- Goleman, D. (1996). *Inteligencia Emocional*. Barcelona: Kairós.
- Greenleaf, R. K. (2003). *Motion and Emotion*. Obtenido de Principal Leadership: <http://www.principals.org/portals/0/content/46875.pdf>
- Hernandez, J. (2003). *Proyecto DAMMAD, Diseño y Aplicación de Modelos Multiagente para Ayuda a la Decisión, TIC2000-1370-C04-02*. Obtenido de Grupo de Investigación de

Sistemas Inteligentes. Universidad Politécnica de Madrid:
<http://platon.escet.urjc.es/grupo/proyectos/dammad/propio/Documentos/Informes/ArquitecturasAgente.doc>

Iglesias, C. (1998). Obtenido de <http://www.upv.es/sma/teoria/agentes/tesiscif.pdf>

Iglesias, C. (1998). *Definición de una Metodología para el Desarrollo de Sistemas Multiagente (Tesis Doctoral)*. Obtenido de Universidad Politécnica de Madrid:
<http://www.upv.es/sma/teoria/agentes/tesiscif.pdf>

Iglesias, C., Garijo, M., & González, J. (1999).

Imbert, R. (2005). *Una Arquitectura Cognitiva Multinivel para Agentes con Comportamiento Influido por Características Individuales y Emociones, Propias y de Otros Agentes*. Obtenido de Archivo Digital UPM-Biblioteca Universitaria-Universidad Politécnica de Madrid: http://oa.upm.es/477/1/Ricardo_Imbert.pdf

Kogan, P., & Roger, S. (2011). *Universidad Nacional de La Plata (UNLP). Grupo de Investigación en Lenguajes e Inteligencia Artificial*. Obtenido de Análisis de opinión como un sistema mutliagente distribuido:
http://sedici.unlp.edu.ar/bitstream/handle/10915/19999/Documento_completo.pdf?sequence=1

Kort, B., Reilly, R., & Picard, R. W. (2001). *An Affective Model of Interplay Between Emotions and Learning: Reengineering Educational Pedagogy - Building a Learning Companion*. Obtenido de <http://affect.media.mit.edu/projectpages/lc/icalt.pdf>

Leshed, G., & Kaye, J. (22-27 de 04 de 2006). *Understanding How Bloggers Feel: Recognizing Affect in Blog Posts*. Obtenido de <http://alumni.media.mit.edu/~jofish/writing/recognizing-bloggers-affect.pdf>

Leshed, G., & Kaye, J. (2006). *Understanding How Bloggers Feel: Recognizing Affect in Blog Posts*. Obtenido de MIT Media Lab:
<http://alumni.media.mit.edu/~jofish/writing/recognizing-bloggers-affect.pdf>

Liu, H., Lieberman, H., & Selker, T. (2003). *A Model of Textual Affect Sensing using Real-World Knowledge*. Obtenido de Larifari.org: http://pop.larifari.org/_/writing/IUI2003-AffectSensing.pdf

Maes, P. (1990). *Designing Autonomous Agents* (Vol. 6). Amsterdam, Netherlands: Elsevier Science Publishers B. V.

- Martín, I., Serrano, Á., Conde, C., & Cabello, E. (2006). Técnicas de Reconocimiento Automático de Emociones. *Revista Electrónica Teoría de la Educación. Educación y Cultura en la Sociedad de la Información.*, 107-127.
- Martín, M. (2012). *Sistema de Clasificación Automática de Críticas de Cine (Proyecto de Fin de Carrera)*. Obtenido de Universidad Carlos III de Madrid: http://e-archivo.uc3m.es/bitstream/10016/5846/1/PFC_Miriam_Martin_Garcia.pdf
- Martínez, E., Martín, M., Perea, J., & Ureña, L. (2011). *Técnicas de clasificación de opiniones aplicadas a un corpus en español*. Obtenido de RUA. Repositorio Institucional de la Universidad de Alicante: http://rua.ua.es/dspace/bitstream/10045/18524/1/PLN_47_17.pdf
- McQuiggan, S. W., Lee, S., & Lester, J. C. (2007). *Early Prediction of Student Frustration*. Obtenido de Department of Computer Science, North Carolina State University, Raleigh, NC 27695: <http://www.intellimedia.ncsu.edu/papers/frustration-acii-2007.pdf>
- Message. (2000).
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
- Nwana, H., Ndumu, D., Lee, L., & Collis, J. (1999). *ZEUS: A Toolkit for Building Distributed Multi-Agent Systems*. Obtenido de CiteSeerX: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.277&rep=rep1&type=pdf>
- Oatley, K. (1992). *Best Laid Schemes: The Psychology of Emotions*. Cambridge University: Maison des Sciences de l'Home.
- Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up? Sentiment Classification using Machine Learning Techniques*. Obtenido de Cornell University-Department of Computer Science: <http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf>
- Pekrun, R. (2006). *The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice*. Obtenido de http://www.scipie.net/docs/2011/area2/Pekrun_EPR_2006.pdf
- Perozo, N. (2011). *Modelado Multiagente para Sistemas Emergentes y Auto-Organizados (Tesis doctoral)*. Obtenido de Universidad de los Andes- Facultad de Ingeniería Doctorado en Ciencias Aplicadas: <http://www.ing.ula.ve/~oteran/tesisPerozo.pdf>
- Rebollo, M. Á., García, R., Barragán, R., Buzón, O., & Vega, L. (2008). Las Emociones en el Aprendizaje Online. *RELIEVE*, v.14, n. 1, p. 2.

- Rizo, R., Llorens, F., & Pujol, M. (2003). *Universidad de Alicante, Dpto. de Ciencia de la Computación e Inteligencia Artificial*. Obtenido de Arquitecturas y Comunicación entre Agentes: <http://www.dccia.ua.es/dccia/inf/asignaturas/AI/docs/ACA.pdf>
- Schröder, M., Baggia, P., Burkhardt, F., Pelachaud, C., Peter, C., & Zovato, E. (2011). *EmotionML- an upcoming standard for representing emotions and related states*. Obtenido de Deutsches Forschungszentrum für Künstliche Intelligenz: http://www.dfki.de/~schroed/articles/schroeder_etal2011b.pdf
- Sebe, N., Sun, Y., Bakker, E., Lew, M., Cohen, I., & Huang, T. (s.f.). *Towards Authentic Emotion Recognition*. Obtenido de CiteSeerX: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=E91EA6EAA9965AEE62201764D7741230?doi=10.1.1.107.4058&rep=rep1&type=pdf>
- Seligman, M., Walker, E., & Rosenhan, D. (2001). *Abnormal psychology* (4th ed.). New York: W. W. Norton & Company, Inc.
- Téllez, A. (2005). *Extracción de Información con Algoritmos de Clasificación (Tesis Maestría)*. Obtenido de INAOE (Instituto Nacional de Astrofísica, Óptica y Electrónica): <http://ccc.inaoep.mx/~mmontesg/tesis%20estudiantes/TesisMaestría-AlbertoTellez.pdf>
- Turney, P. (2002). *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. Philadelphia: 40th Annual Meeting of the Association for Computational Linguistics (ACL). Obtenido de <http://acl.ldc.upenn.edu/P/P02/P02-1053.pdf>
- Turney, P., & Littman, M. (2003). *Measuring Praise and Criticism: Inference of Semantic Orientation from Association*. Obtenido de Cogprints: <http://cogprints.org/3164/1/turney-littman-acm.pdf>
- Vapnik, V. (1999). *An Overview of Statistical Learning Theory*. Obtenido de Massachusetts Institute of Technology: http://web.mit.edu/6.962/www/www_spring_2001/emin/slt.pdf
- Wesson, R., & Hayes-Roth, F. (1980). *RAND CORPORATION*. Obtenido de Network Structures for Distributed Situation Assessment: <http://www.rand.org/content/dam/rand/pubs/reports/2006/R2560.pdf>
- Wooldridge, M., Jennings, N., & Kinny, D. (2000). *The Gaia Methodology for Agent-Oriented Analysis and Design*. Obtenido de University of Southampton: <http://eprints.soton.ac.uk/253748/1/jaamas2000.pdf>

ANEXOS

ANEXO 1: Mensajería

ANEXO1-A: Mensajes utilizados para las pruebas

Estimada profesora he revisado el cuadro de notas y me encuentro con que estoy reprobado por no entregar trabajos,pero si tengo nota en el campo de trabajo, no entiendo y que deberia hacer !!!! Gracias

Gracias ya lo he revisado y le estoy muy agradecida por la atención al presente. Saludos Cordiales,

Estimada Ing. Por pedirle un favor, El link del libro : ""Ontological Engineering"" que se publico en el eva(Tema 6: Materia de Sistemas Basados en el Conocimiento) esta roto por lo que no es posible descargarlo. Por favor, ayudenos arreglando el link de descarga. Agradeciendole la atención prestada al presente,

Estimada Profesora, respecto al ejercicio del módulo 7, no entiendo que no más se debe hacer ¿solo escrit para crear la BDD y las tablas o también formularios para manipular registros de las tanlas? Además, ¿en que se diferencia esto del trabajo final? Muchas gracias por su ayuda. Estimada Maestra.

La actividad del módulo 7 dice: ""...siguientes tablas: Producto, Proveedor, Cliente, Detalle...proceso de crear la base de datos y ... tablas debe desde rutinas php."" Mi pregunta es ¿que hace la tabla detalle? ¿es detalle de factura o pedido? Si es así ¿podemos crear la tabla ""maestra"" factura o pedido? Saludos, Gracias por su respuesta.

Saludos, muchas gracais Ing por su ayuda espero esten bien los trabajo de ensayo ya que fui el unico que participe en el foro y todo lo pedido por usted muchas gracias

Ok, muchas gracias por sus observaciones.

Estimada Ingeniera Por favor quiero solicitarle nos ayude con un manual de Hot Potatoes para Mac, ya que he descargado el programa para Mac pero observo que es diferente a lo que se reviso en la capacitación. Gracias Ingeniera, estaré pendiente.

Estimada Ingeniera, reciba un cordial y atento saludo. Por favor si fuera tan amable de

informarnos que actividades debemos emprender en el Seminario II. Gracias por su información.

gracias... Ingeniera... por su ayuda

Ingeniera buenas noches, queria pedirle de favor especial si me puede ayudar para entregarle la pagina web hasta el dia lunes ya que por motivos de trabajo llevo ya 2 semanas en Quito y no tengo las herramientas para poder trabajar, En espera de su respuesta desde ya le anticipo mis sinceros agradecimientos. Saludos

Buenas tardes por favor su ayuda con un guía referente a la pregunta explique los modelos de síntomas dependientes y de síntomas independientes.

Muchas gracias le comento que realice el procedimiento indicado clasificador pulsando el botón Choose. Aparecerá una estructura de directorios en la que se seleccionará el directorio trees y dentro del él el algoritmo J48. Pero no aparece el diagrama como se detalla en el pdf, por favor su ayuda

Por favor su ayuda, la opción de visualizar el árbol de una manera más atractiva si pulsamos el botón derecho sobre el texto caja Result-list. Seleccionamos la opción Visualize Tree, y obtendremos el árbol no me funciona le comento dispongo la versión Weka 3.6.4.

Le comento que acabo de regresar del centro y aun no estan entregando los exámenes, por lo cual queria molestarle para ver si es posible si me ayuda con la evaluacion, esto para poder es revisar y correguir en miras para el supletorio. De antemano mil gracias.

Molesto su atencion para solicitarle nos ayude con el examen correguido, pues para el supletorio seria de gran ayuda conocer en que fallamos. De antemano muchas gracias"

Muchas Gracias, quisiera consultarle sobre que tipo de examen sera, de los de cuadernillo de preguntas o los de ejercicios? y si no es abuso como que temas se evaluara, es decir: conceptos, codigos, ejercicios etc?? De antemano mil gracias por su ayuda.

Quería consultarle si el fin de semana tendremos que rendir una evaluación presencial de la materia Seminario II??

Mil gracias Ing. por su ayuda. Compañeros que estan en el listado, ya les habra llegado la

invitación, porque a mi aun no.

Compañeros para dropbox deben registrarse y luego crear una carpeta llamada practicumI_nombre y apellido, dentro crear la carpeta google maps colocar la foto y video,

Buenas tardes. También tengo problemas tratando de añadir la marca en el mapa. Siguiendo con lo anterior, luego hay que compartir la carpeta practicumI_nombre con la Ing. al correo.

Buenas noches ingeniera seria tan amable de darnos un poco mas de tiempo para entregar los mapas conceptuales, son 2 y solo tenemos una semana...Espero pueda ayudarnos..Gracias

Buenos dias Ing. Por favor nos podría indicar las preguntas de Autoevaluación de Repaso las subimos al EVA, y como podemos subirlas. Mil gracias

Buenas noches ingeniera seria tan amable de darnos un poco mas de tiempo para entregar los mapas conceptuales, son 2 y solo tenemos una semana... Espero pueda ayudarnos.. Gracias

Por cierto y la nota del 1B???. Como se que estoy bien en su materia???

Estimada profesora, podría indicarme cual es el contexto del poster? la empresa que analizamos o la situación de las TICs en la ciudad?

Ingeniera nos podría decir a donde subimos el informe....o en donde esta el enlace???

No puedo insertar el video... no se que pasa, alguien puede ayudarme?

Estimada profesora si es posible que nos oriente con metodología UML?

Buenas tardes Ingeniera. ¿Cuándo podremos saber las notas del primer bimestres?

ANEXO2-A: Código de implementación de Agentes

✓ Agente Extracción

```
public class Extraccion extends Agent{

public Extraccion() {
    }

    private boolean recoleccionDatos(){
        //Conexion a la base de datos
return false;
    }

    protected void setup() {
if(recoleccionDatos()){
        System.out.println("Datos recolectados");
    }else{
        System.out.println("Error en la recoleccion de Datos
recolectados");
    }
    }

}
}
```

✓ Agente PLN

```
public class Pln extends Agent {

String []
stopwords={"él", "ésta", "ésta", "éste", "éstos", "última", "últimas", "último", "últimos",
"a", "añadió", "aún", "actualmente", "adelante", "además", "afirmó", "agregó", "ahí", "ahor
a", "al", "algún",
"algo", "alguna", "algunas", "alguno", "algunos", "alrededor", "ambos", "ante", "anterior",
"antes", "apenas", "aproximadamente", "aquí", "así", "aseguró", "aunque", "ayer", "bajo", "b
ien", "buen", "buena", "buenas", "bueno", "buenos", "cómo", "cada", "casi", "cerca", "cierto",
"cinco", "comentó", "como", "con", "conocer", "consideró", "considera", "contra", "cosas",
"creo", "cual", "cuales", "cualquier", "cuando", "cuanto", "cuatro", "cuenta", "da", "dado",
"dan", "dar", "de", "debe", "deben", "debido", "decir", "dejó", "del", "demás", "dentro", "des
de", "después", "dice", "dicen", "dicho", "dieron", "diferente", "diferentes", "dijeron", "d
ijo", "dio", "donde", "dos", "durante", "e", "ejemplo", "el", "ella", "ellas", "ello", "ellos",
"embargo", "en", "encuentra", "entonces", "entre", "era", "eran", "es", "esa", "esas", "ese",
"eso", "esos", "está", "están", "esta", "estaba", "estaban", "estamos", "estar", "estará", "
estas", "este", "esto", "estos", "estoy", "estuvo", "ex", "existe", "existen", "explicó", "ex
presó", "fin", "fue", "fuera", "fueron", "gran", "grandes", "ha", "había", "habían", "haber",
"habrá", "hace", "hacen", "hacer", "hacerlo", "hacia", "haciendo", "han", "hasta", "hay", "ha
ya", "he", "hecho", "hemos", "hicieron", "hizo", "hola", "hoy", "hubo", "igual", "incluso", "i
ndicó", "informó", "ing.", "junto", "la", "lado", "las", "le", "les", "llegó", "lleva", "lleva
r", "lo", "los", "luego", "lugar", "más", "manera", "manifestó", "mayor", "me", "mediante", "m
ejor", "mencionó", "menos", "mi", "mientras", "misma", "mismas", "mismo", "mismos", "momento",
"mucho", "muchas", "mucho", "muchos", "muy", "nada", "nadie", "ni", "ningún", "ninguna", "n
ingunas", "ninguno", "ningunos", "no", "nos", "nosotras", "nosotros", "nuestra", "nuestras",
"nuestro", "nuestros", "nueva", "nuevas", "nuevo", "nuevos", "nunca", "o", "ocho", "otra", "
otras", "otro", "otros", "para", "parece", "parte", "partir", "pasada", "pasado", "pero", "pe
```

```

sar", "poca", "pocas", "poco", "pocos", "podemos", "podrá", "podrán", "podría", "podrían", "p
oner", "por", "porque", "posible", "próximo", "próximos", "primer", "primera", "primero", "p
rimeros", "principalmente", "propia", "propias", "propio", "propios", "pudo", "pueda", "pue
de", "pueden", "pues", "qué", "que", "quedó", "queremos", "quién", "quien", "quienes", "quier
e", "realizó", "realizado", "realizar", "respecto", "sí", "sólo", "se", "señaló", "sea", "sea
n", "según", "segunda", "segundo", "seis", "ser", "será", "serán", "sería", "si", "sido", "sie
mpre", "siendo", "siete", "sigue", "siguiente", "sin", "sino", "sobre", "sola", "solamente",
"solas", "solo", "solos", "son", "su", "sus", "tal", "también", "tampoco", "tan", "tanto", "te
nía", "tendrá", "tendrán", "tenemos", "tener", "tenga", "tengo", "tenido", "tercera", "tiene
", "tienen", "toda", "todas", "todavía", "todo", "todos", "total", "tras", "trata", "través",
"tres", "tuvo", "un", "una", "unas", "uno", "unos", "usted", "va", "vamos", "van", "varias", "v
arios", "veces", "ver", "vez", "vosotros", "y", "ya", "yo", "pero", "yo"};

```

```

private ArrayList<String> eliminarPalabrasIrrelevantes(ArrayList<String> mensaje) {
    ArrayList<String> limpio=mensaje;
    for (int i = 0; i < stopwords.length; i++) {
        mensaje.remove(stopwords[i]);
    }
    return limpio;
}

```

```

protected void setup() {
    System.out.println("Iniciando Agente-PLN");
}

```

```

//Limpiar documento resultados
    Utilidad.limpiarArchivo("resultado_pln.txt");
//Abrir mensaje
    BufferedReader br = null;
    try {
        br = new BufferedReader(new FileReader(Utilidad.ruta+ "mensajes.txt"));
        String output;
        while ((output = br.readLine()) != null) {
String mensaje=output+" - -";
            //Tokenizar mensaje
            ArrayList<String> linea = Utilidad.tokenizarLinea(mensaje);
            //Eliminar palabras irrelevantes
            linea = eliminarPalabrasIrrelevantes(linea);
            //Guardar en resultado_pln.txt
            Utilidad.guardarResultado("resultado_pln.txt", linea);
        }
    } catch (FileNotFoundException ex) {
        Logger.getLogger(Extraccion.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (IOException ex) {
        Logger.getLogger(Extraccion.class.getName()).log(Level.SEVERE, null,
ex);
    } finally {
        try {
            br.close();
        } catch (IOException ex) {
            Logger.getLogger(Extraccion.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
    //Abrir mensaje
    System.out.println("FIN Agente-PLN");
}
}

```

✓ Agente Polaridad

```
public class Polaridad extends Agent {

    HashMap diccionarioSinonimos;
    String respuestal="";

    private void cargarDiccionario(String ruta_archivo) {
        diccionarioSinonimos = new HashMap();
        BufferedReader br = null;
        boolean aux1 = false;
        try {
            br = new BufferedReader(new FileReader(ruta_archivo));
            String output;
            while ((output = br.readLine()) != null) {
                String[] aux = output.split("\t");
                for (int i = aux.length - 1; i > 0; i--) {
                    diccionarioSinonimos.put(aux[i], aux[0]);
                }
            }
        } catch (FileNotFoundException ex) {
            Logger.getLogger(Extraccion.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (IOException ex) {
            Logger.getLogger(Extraccion.class.getName()).log(Level.SEVERE, null,
ex);
        } finally {
            try {
                br.close();
            } catch (IOException ex) {
                Logger.getLogger(Extraccion.class.getName()).log(Level.SEVERE, null,
ex);
            }
        }
    }

    private String[][] trigram(ArrayList<String> secuencia) {

        String[][] trigrams = new String[secuencia.size() - 2][];
        for (int i = 0; i < trigrams.length; i++) {
            String[] aux = {secuencia.get(i), secuencia.get(i + 1), secuencia.get(i +
2)};
            trigrams[i] = aux;
        }

        return trigrams;
    }

    private ArrayList<String> determinarEmociones() {
        ArrayList<String> emociones = new ArrayList<String>();
        String var1 = "";
        String var2 = "";
        String var3 = "";
        String var4 = "";
        String var5 = "";

        //=====Cargar Archivos necesarios=====
        cargarDiccionario(Utilidad.ruta + "raices.txt");//Cargar diccionario
        ArrayList<String> negacion = Utilidad.tokenizarArchivo("negacion.txt");
        ArrayList<String> confusion =
Utilidad.tokenizarArchivo("confusionTri.txt");
        ArrayList<String>frustración =Utilidad.tokenizarArchivo("frustracionTri.txt");
        ArrayList<String> verbos = Utilidad.tokenizarArchivo("verbos2.txt");
        ArrayList<String> raices = Utilidad.tokenizarArchivo("raices.txt");
    }
}
```



```

ArrayList<String> intensificadores =
Utilidad.tokenizarArchivo("Adv_intensificadores.txt");

//=====Cargar Archivos necesarios=====

//=====Abrir archivo de mensajes limpios
(resultado_pln.txt)=====
    int globalAnsiedad = 0;
    int globalAburrimiento = 0;
    int globalConfusion = 0;
    int globalFrustracion = 0;
    int globalSimpatia = 0;
    int globalAngustia = 0;
    BufferedReader br = null;
    String resultado_analisis = "";
    try {
        br = new BufferedReader(new FileReader(Utilidad.ruta +
"resultado_pln.txt"));
String output;
        int contadorMensajes = 0;
        int etiquetadoCorrecto = 0;
        int noEtiquetado = 0;
        int error = 0;
        while ((output = br.readLine()) != null) {

contadorMensajes++;

        System.out.println("=====MENSAJE:" + contadorMensajes +
"=====");
ArrayList<String> tokens = Utilidad.tokenizarLinea(output);
        System.out.println("Mensaje:" + output);
        System.out.println("Tokens:" + tokens);

        //Crear trigrams
        String[][] trig = trigram(tokens);
        System.out.println("Trigram Tamano: " + trig.length);

int verificaEtiqueta = 0;
        int suma = 0;

        //=====Inicializar contadores de emocion
            int contadorConfusion = 0;
            int contadorAngustia = 0;
            int contadorAburrimiento = 0;
            int contadorSimpatia = 0;
            int contadorAnsiedad = 0;
            int contadorFrustracion = 0;

        //=====Inicializar contadores de emocion

            for (int var = 0; var < trig.length; var++) {
if (var < trig.length) {
                int salto = 0;
                if (salto == 0) {
                    for (int a = 0; a < verbos.size(); a++) {
                        if (verbos.get(a).equals(trig[var][0])) {
                            for (int x = 0; x < intensificadores.size();
x++) {
                                if (intensificadores.get(x).equals(trig[var][1])) {
#Raices de sinonimos
                                    for (int b = 0; b < raices.size(); b++) {
if (trig[var][0].equals(verbos.get(a))
&&(trig[var][1].equals(intensificadores.get(x))) &&

```



```

contadorAnsiedad++;
}
if (diccionarioSinonimos.get(raices.get(b)).equals("frustración")) {
contadorFrustracion++;
}
}
}

if (salto == 0) {
for (int c = 0; c < confusion.size(); c++) {
if (salto == 0) {
if (trig[var][2].startsWith(confusion.get(c))) {

System.out.println("TRIGRAM ENCONTRADO");

System.out.println(Arrays.toString(trig[var]));
System.out.println("CONFUSION ENCONTRADA");
salto = 1;
contadorConfusion++;
}
if (trig[var][2].equals("sé") || trig[var][2].equals("se"))
{System.out.println("TRIGRAM ENCONTRADO");

System.out.println(Arrays.toString(trig[var]));
System.out.println("CONFUSION ENCONTRADA");
salto = 1;
contadorConfusion++;
}
}
}

if (salto == 0) {
for (int d = 0; d < frustracion.size(); d++) {
if (trig[var][2].startsWith(frustracion.get(d))) {
System.out.println("TRIGRAM ENCONTRADO");
System.out.println(Arrays.toString(trig[var]));
System.out.println("FRUSTRACION ENCONTRADA");
salto = 1;

contadorFrustracion++;
}
}
}

if (salto == 0) {
for (int b = 0; b < frustracion.size(); b++) {
if (trig[var][1].startsWith(frustracion.get(b))) {
System.out.println("Bigram Encontrado");
System.out.println(trig[var][0] + ", " + trig[var][1]);
System.out.println("EMOCION DESCRITA:");
System.out.println("FRUSTRACION:");

salto = 1;

contadorFrustracion++;
}
}
}

if (salto == 0) {
for (int c = 0; c < confusion.size(); c++) {
if (trig[var][1].startsWith(confusion.get(c))) {
System.out.println("Bigram Encontrado");
System.out.println(trig[var][0] + ", " +
trig[var][1]);
System.out.println("EMOCION DESCRITA:");
System.out.println("CONFUSION:");
salto = 1;
contadorConfusion++;
}
}
}
}
}
}

```

```

    }
    if (salto == 0) {
        if (trig[var][1].equals("sé") || trig[var][1].equals("se")) {
            System.out.println("BIGRAM ENCONTRADO");
            System.out.println(trig[var][0] + ", " + trig[var][1]);
        }
        System.out.println("CONFUSION ENCONTRADA");
        salto = 1;
        contadorConfusion++;
    }
}
//
        if (salto == 0) {
for (int d = 0; d < raices.size(); d++) {
            if (trig[var][1].startsWith(raices.get(d))) {
                System.out.println("MENSAJE");
                System.out.println(trig[var][0] + ", " + trig[var][1]);
                System.out.println("BIGRAM ENCONTRADA");
                salto = 1;
                break;
            }
        }
    }
}

if (salto == 0) {
for (int a = 0; a < verbos.size(); a++) {
        if (salto == 0) {
if (verbos.get(a).equals(trig[var][0])) {
for (int b = 0; b < raices.size(); b++) {
                if (trig[var][0].equals(verbos.get(a)) &&
(trig[var][1].startsWith(raices.get(b)) && salto == 0)) {
                    System.out.println("BIGRAM");
                    System.out.println(trig[var][0] + ", " +
trig[var][1]);
                    System.out.println(diccionarioSinonimos.get(raices.get(b)));
                    System.out.println("ENCONTRADA");
                    salto = 1;
                }

if (diccionarioSinonimos.get(raices.get(b)).equals("angustia")) {
                    contadorAngustia++;
                }

                if
(diccionarioSinonimos.get(raices.get(b)).equals("confusión")) {
                    contadorConfusion++;
                }

if (diccionarioSinonimos.get(raices.get(b)).equals("aburrimiento"))
{
                    contadorAburrimiento++;
                }

                if (diccionarioSinonimos.get(raices.get(b)).equals("simpatia"))
{
                    contadorSimpatia++;
                }

                if (diccionarioSinonimos.get(raices.get(b)).equals("ansiedad"))
{
                    contadorAnsiedad++;
                }

                if
(diccionarioSinonimos.get(raices.get(b)).equals("frustración")) {
                    contadorFrustracion++;
                }

                break;
            }
        }
    }
}

```



```

        }
        if (trig[var][2].startsWith(raices.get(b))) {
            System.out.println("UNIGRAM");
            System.out.println(trig[var][2]);
            System.out.println(diccionarioSinonimos.get(raices.get(b)));
            System.out.println("ENCONTRADA");
            salto = 1;
            if (diccionarioSinonimos.get(raices.get(b)).equals("angustia")) {
                contadorAngustia++;
            }
            if (diccionarioSinonimos.get(raices.get(b)).equals("confusión"))
        {
            contadorConfusion++;
        }
            if
        (diccionarioSinonimos.get(raices.get(b)).equals("aburrimiento")) {
                contadorAburrimiento++;
            }
            if (diccionarioSinonimos.get(raices.get(b)).equals("simpatia")) {
                contadorSimpatia++;
            }
            if (diccionarioSinonimos.get(raices.get(b)).equals("ansiedad")) {
                contadorAnsiedad++;
            }
            if
        (diccionarioSinonimos.get(raices.get(b)).equals("frustración")) {
                contadorFrustracion++;
            }
        }
    }
}

suma = contadorConfusion + contadorAngustia + contadorAburrimiento +
contadorSimpatia + contadorAnsiedad + contadorFrustracion;

//Presentacion de Dato
if (suma != 0) {
    System.out.println("CARGA EMOCIONAL DEL MENSAJE");
    var1 = "SIMPATIA: " + ((contadorSimpatia * 100) / suma);
    System.out.println(var1);
    var2 = "CONFUSION: " + ((contadorConfusion * 100) / suma);
    System.out.println(var2);
    var3 = "ANGUSTIA: " + ((contadorAngustia * 100) / suma);
    System.out.println(var3);
    var4 = "ANSIEDAD: " + ((contadorAnsiedad * 100) / suma);
    System.out.println(var4);
    var5 = "FRUSTRACION: " + ((contadorFrustracion * 100) /
suma);
    System.out.println(var5);

    if (contadorAnsiedad > contadorSimpatia && contadorAnsiedad
> 0 && contadorAnsiedad > contadorAngustia && contadorAnsiedad >
contadorFrustracion && contadorAnsiedad > contadorConfusion && contadorAnsiedad
> contadorAburrimiento) {
        System.out.println("Polaridad del mensaje: (-)
ANSIEDAD");
        verificaEtiqueta = 1;
        globalAnsiedad++;
    }
    if (contadorFrustracion > contadorSimpatia &&

```

```

contadorFrustracion > 0 && contadorFrustracion > contadorAngustia &&
contadorFrustracion > contadorAnsiedad && contadorFrustracion >
contadorConfusion && contadorFrustracion > contadorAburrimiento) {
    System.out.println("Polaridad del Mensaje: (-)
FRUSTRACIÓN");
    verificaEtiqueta = 1;
    globalFrustracion++;
}
if (contadorAngustia > contadorSimpatia && contadorAngustia
> 0 && contadorAngustia > contadorFrustracion && contadorAngustia >
contadorAnsiedad && contadorAngustia > contadorConfusion && contadorAngustia >
contadorAburrimiento) {
    System.out.println("Polaridad del Mensaje: (-)
ANGUSTIA");
verificaEtiqueta = 1;
    globalAngustia++;
}
if (contadorConfusion > contadorSimpatia &&
contadorConfusion > 0 && contadorConfusion > contadorFrustracion &&
contadorConfusion > contadorAnsiedad && contadorConfusion > contadorAngustia &&
contadorConfusion > contadorAburrimiento) {
    System.out.println("Polaridad del Mensaje: (-)
CONFUSION");
    verificaEtiqueta = 1;
    globalConfusion++;
}
if (contadorSimpatia > contadorFrustracion &&
contadorSimpatia > contadorAnsiedad && contadorSimpatia > contadorAngustia &&
contadorSimpatia > contadorAburrimiento && contadorSimpatia > contadorConfusion)
{
    System.out.println("Polaridad del Mensaje: (+)
SIMPATIA");
    verificaEtiqueta = 1;
    globalSimpatia++;
}
}

if (suma > 0 && verificaEtiqueta == 0) {
    error++;
    System.out.println("Error: " + error);
}
if (suma > 0 && verificaEtiqueta == 1) {
    etiquetadoCorrecto++;
}
if (suma == 0 && verificaEtiqueta == 0) {
    noEtiquetado++;
}

//Guardar los resultados
ArrayList<String> resultado = new ArrayList<String>();
boolean valor = false;
if (contadorSimpatia > 0) {
    valor = true;
    valor = true;
    resultado.add(var1);
    resultado_analisis = var1;
}
if (contadorConfusion > 0) {
    resultado.add(var2);
    valor = true;
    resultado_analisis = var2;
}
}

```

```

        if (contadorAngustia > 0) {
            resultado.add(var3);
            valor = true;
            resultado_analisis = var3;
        }
        if (contadorAnsiedad > 0) {
            resultado.add(var4);
            valor = true;
            resultado_analisis = var4;
        }
        if (contadorFrustracion > 0) {
            resultado.add(var5);
            valor = true;
            resultado_analisis = var5;
        }
        if (valor) {
respuestal=resultado_analisis;
            Utilidad.guardarResultado("resultado_polaridad.txt",
resultado);
        }

System.out.println("=====");
        }
        emociones.add(resultado_analisis);
    } catch (FileNotFoundException ex) {
        System.out.println("Error: " + ex.getMessage());
    } catch (IOException ex) {
        System.out.println("Error: " + ex.getMessage());
    } finally {
        try {
            br.close();
        } catch (IOException ex) {
            System.out.println("Error: " + ex.getMessage());
        }
    }
}

//=====Abrir archivo de mensajes limpios
(resultado_pnl.txt)=====
System.out.println("=====VARIABLES=====");
System.out.println("variable 1:" + var1);
System.out.println("variable 2:" + var2);
System.out.println("variable 3:" + var3);
System.out.println("variable 4:" + var4);
System.out.println("variable 5:" + var5);
System.out.println("=====");
return emociones;
}

private Codec codec = new SLCodec();
private Ontology ontologia = agentes.ontologia.Ontologia.getInstance();

class EnviarMensajeBehaviour extends SimpleBehaviour {

    private boolean finished = false;

    public EnviarMensajeBehaviour(Agent a) {
        super(a);
    }

    public void action() {
        try {
String respuesta = "Ontologia";
            AID r = new AID();
            r.setLocalName(respuesta);

```



```

        ACLMessage msg = new ACLMessage (ACLMessage.INFORM);
        msg.setSender (getAID());
        msg.addReceiver (r);
        msg.setLanguage (codec.getName());
        msg.setOntology (ontologia.getName());
    agentes.ontologia.Tipo tipo = new agentes.ontologia.Tipo();
    tipo.setTipoEmocion (respuestal);
    agentes.ontologia.Polaridad polaridad = new agentes.ontologia.Polaridad();
        polaridad.setTipoPolaridad (tipo);

        getContentManager().fillContent (msg, polaridad);

        send (msg);

    } catch (jade.content.lang.Codec.CodecException ce) {
System.out.println ("error"+ce.getMessage());
        System.out.println (ce);
    } catch (jade.content.onto.OntologyException oe) {
        System.out.println ("Error1"+oe);
    }
    catch (Exception e) {
System.out.println ("\n\n... Terminando ...");
        finished = true;
    }
    }
    public boolean done() {

        return finished;

    }
}

protected void setup() {
System.out.println ("Iniciando Agente-Polaridad...");

    //Limpiar documento resultados
    Utilidad.limpiarArchivo ("resultado_polaridad.txt");

    ArrayList<String> emociones = determinarEmociones();
    System.out.println ("Emociones: " + emociones);
System.out.println ("Fin polaridad.");
}

DFAgentDescription dfd = new DFAgentDescription();
    ServiceDescription sd = new ServiceDescription();
sd.setType ("Vendedor");
    sd.setName (getName());
    sd.setOwnership ("ARNOIA");
    dfd.setName (getAID());
    dfd.addServices (sd);
    try {
        DFService.register (this, dfd);
    } catch (FIPAException e) {
System.err.println (getLocalName() + " registration with DF unsucceeded. Reason:
"+e.getMessage());
        doDelete();
    }

    getContentManager().registerLanguage (codec);
    getContentManager().registerOntology (ontologia);

    EnviarMensajeBehaviour EnviarBehaviour = new EnviarMensajeBehaviour (this);
    addBehaviour (EnviarBehaviour);

}

```

```

protected void takeDown() {
    try {
        DFService.deregister(this);
    } catch (FIPAException fe) {
        fe.printStackTrace();
    }
}
}

```

✓ Agente Ontología

```

public class Ontologia extends Agent {
    private Codec codec = new SLCodec();
    private Ontology ontologia = agentes.ontologia.Ontologia.getInstance();

    // Clase que describe el comportamiento que permite recibir un mensaje
    // y contestarlo
    class WaitPingAndReplyBehaviour extends SimpleBehaviour {
        private boolean finished = false;

        public WaitPingAndReplyBehaviour(Agent a) {
            super(a);
        }

        public void action() {

            System.out.println("\nEsperando tipo de emocion...");

            MessageTemplate mt = MessageTemplate.and(
                MessageTemplate.MatchLanguage(codec.getName()),
                MessageTemplate.MatchOntology(ontologia.getName()));
            ACLMessage msg = blockingReceive(mt);

            try {

                if(msg != null){
                    if(msg.getPerformative() == ACLMessage.NOT_UNDERSTOOD){
                        System.out.println("Mensaje recibido no entendido");
                    }
                    else{
                        if(msg.getPerformative() == ACLMessage.INFORM) {

                            ContentElement ce = getContentManager().extractContent(msg);
                            if (ce instanceof agentes.ontologia.Polaridad){

                                // Recibido un INFORM con contenido correcto
                                agentes.ontologia.Polaridad polaridad = (agentes.ontologia.Polaridad)
ce;
                                agentes.ontologia.Tipo tipo = polaridad.getTipoPolaridad();
                                System.out.println("Mensaje recibido:");
                                System.out.println("Emocion: " + tipo.getTipoEmocion());

                                //Pedimos
                                agentes.ontologia.Porcentaje porcentaje = new
                                agentes.ontologia.Porcentaje();
                                porcentaje.setTotalPorcentaje(tipo);
                                ACLMessage msg2 = new ACLMessage(ACLMessage.REQUEST);
                                msg2.setLanguage(codec.getName());
                                msg2.setOntology(ontologia.getName());
                                msg2.setSender(getAID());

```

```

        msg2.addReceiver(msg.getSender());
        getContentManager().fillContent(msg2,porcentaje);
        send(msg2);
    }

    else{
        // Recibido un INFORM con contenido incorrecto
        ACLMessage reply = msg.createReply();
        reply.setPerformative(ACLMessage.NOT_UNDERSTOOD);
        reply.setContent("( UnexpectedContent (expected ping) )");
        send(reply);
    }
}

else {

    // Recibida una performativa incorrecta
    ACLMessage reply = msg.createReply();
    reply.setPerformative(ACLMessage.NOT_UNDERSTOOD);
    reply.setContent("( Unexpected-act
"+ACLMessage.getPerformative(msg.getPerformative())+" ( expected (inform)) )");
    send(reply);
}

}
}else{
//System.out.println("No message received");
}

}

catch (jade.content.lang.Codec.CodecException ce){
    System.out.println(ce);
}

catch (jade.content.onto.OntologyException oe) {
    System.out.println("ErrorOntologia"+oe);
}

}

}

public boolean done() {
    return finished;
}

} //Fin de la clase WaitPingAndReplyBehaviour

protected void setup() {

//    addBehaviour(new WaitPingAndReplyBehaviour(this));
    getContentManager().registerLanguage(codec);
    getContentManager().registerOntology(ontologia);
    WaitPingAndReplyBehaviour PingBehaviour;
    PingBehaviour =new WaitPingAndReplyBehaviour(this);
    addBehaviour(PingBehaviour);
}
}
}

```

✓ Agente Interfaz

```

public class SistemaMutiAgente extends javax.swing.JFrame {
    private Utilidad u = Utilidad.getUtilidad();
    private Interfaz agente;
}

```

```

/**
 * Creates new form SistemaMutiAgente
 */
public SistemaMutiAgente(Interfaz a) {
    initComponents();
    this.agente=a;
}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_jButton1MouseClicked
// TODO add your handling code here:
    this.agente.ordenarAnalisis();
    BufferedReader br = null;

    try {
        br = new BufferedReader(new
FileReader(u.getRuta()+"resultado_polaridad.txt"));
String output;
        int num_mensaje=1;
        while ((output = br.readLine()) != null) {

this.jTextArea1.setText(this.jTextArea1.getText()+"Mensaje"+num_mensaje+":
"+output+"\n");
num_mensaje++;
        }
        } catch (FileNotFoundException ex) {
            System.out.println("Error"+ex.getMessage());
        } catch (IOException ex) {
            System.out.println("Error"+ex.getMessage());
        } finally {
            try {
                br.close();
            } catch (IOException ex) {
                System.out.println("Error"+ex.getMessage());
            }
        }
    }

private void BuscarActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_BuscarActionPerformed
JFileChooser chooser = new JFileChooser();
    chooser.setCurrentDirectory(new File(".txt"));
    javax.swing.filechooser.FileFilter txt = new
FileNameExtensionFilter("Archivo txt", "txt");
chooser.addChoosableFileFilter(txt);
    int status = chooser.showOpenDialog(this);
    if (status == JFileChooser.APPROVE_OPTION) {
File selectedFile = chooser.getSelectedFile();
        txtFile.setText(selectedFile.getAbsolutePath());
        u.setArchivo(selectedFile.getAbsolutePath());
        System.out.println(selectedFile.getParent());
        System.out.println(selectedFile.getName());
    } else if (status == JFileChooser.CANCEL_OPTION) {
        System.out.println(JFileChooser.CANCEL_OPTION);
    }
}
}

```

Arquitectura multiagente para detectar estados emocionales de los estudiantes en un curso virtual

Cabrera Montoya Karla Yolanda
Universidad Técnica Particular de Loja
UTPL
Loja, Ecuador
kycabrera@utpl.edu.ec

Mgs. Valdiviezo Díaz Priscila Marisela
Universidad Técnica Particular de Loja
UTPL
Loja, Ecuador
pmvaldiviezo@utpl.edu.ec

Resumen— En el presente trabajo se ha obtenido un modelo de Arquitectura Multiagente aplicando la metodología MAS-CommonKADS, y con ayuda del análisis de texto de los foros y comentarios de la Red Social del entorno virtual de aprendizaje EVA, de los estudiantes, se ha podido establecer la polaridad emocional de forma no intrusiva que presenta el texto mediante la utilización de técnicas de procesamiento de lenguaje natural, permitiendo así, dar a conocer al docente cuando es necesario proporcionar una retroalimentación a los estudiantes en sus respectivos componentes académicos.

Keywords—Análisis de sentimientos, emociones y aprendizaje, MAS_COMMONKADS, arquitectura multiagente, agentes inteligentes, sistemas informáticos, minería de opinión, análisis de contenido.

1. INTRODUCCIÓN

Ante la gran cantidad de texto con contenido subjetivo, surge el análisis de sentimientos que trata de clasificar el texto según su polaridad, mediante la utilización de técnicas de PLN y minería de texto. Al ser un tema de actualidad tiene mucho potencial para ser explotado al máximo, se lo puede trabajar desde varios puntos, ya sea el desarrollo de aplicaciones de escritorio, web o móviles, o como se lo maneja actualmente en algunas redes sociales.

Por estas razones se cree conveniente el desarrollo de una Arquitectura Multiagente para detectar estados emocionales, en el ámbito estudiantil especialmente en educación a distancia, ya que al no existir una interacción “cara a cara” entre el docente y el estudiante se dificulta su comunicación y el docente no puede brindar una retroalimentación necesaria de los temas con los cuales los estudiantes tiene ciertos problemas de aprendizaje.

Se aplica la metodología MAS_CommonKADS, que está orientada al desarrollo de sistemas multiagente mediante el uso de modelos para la realización de la conceptualización y análisis, mientras que su implementación se realizó independientemente de estos modelos mencionados, ya que esta metodología nos ofrece la funcionalidad de trabajar los requerimientos independientemente del diseño o implementación.

Finalmente, se presenta el modelo de diseño de la Arquitectura multiagente propuesta, un modelo de agentes simple, así como los componentes que conforman dicha arquitectura, su implementación en JADE y sus pruebas correspondientes. Y finalmente las conclusiones y trabajos futuros.

2. METODOLOGÍA APLICADA

Para el desarrollo de este proyecto se ha seleccionado la metodología MAS-CommonKADS (Iglesias, 1998), que se enfoca principalmente en la construcción de sistemas inteligentes, permite la integración de técnicas de ingeniería de conocimientos e ingeniería de software orientada a objetos y de protocolos, para una mejor comprensión de esta metodología se explica en el siguiente apartado una visión general de la metodología seleccionada. La aplicación de esta metodología consiste en el desarrollo de modelos, se define una plantilla textual en la cual se establecen todos los requerimientos que forman parte de la solución.

2.1 CONCEPTUALIZACIÓN

- Comprensión del Problema

Uno de los mayores problemas del aprendizaje en un entorno virtual, es la dificultad para detectar el estado emocional de los estudiantes, impidiendo brindarles retroalimentación para mejorar su motivación y su rendimiento académico.

- Ciclo de Desarrollo de los Casos de Uso

Los casos de uso describen una secuencia de interacciones que se realiza entre un sistema y sus actores. Los actores son externos al sistema.

- ✓ Identificación y Descripción de los Actores

Actores Activos:

- *Estudiante*: interactúa con el docente a través del entorno virtual de aprendizaje (EVA) mediante foros, mensajes, comentarios, videos.
- *Docente*: interactúa con el estudiante a través del entorno virtual de aprendizaje (EVA) mediante tareas, tutoriales, foros, mensajes, comentarios, videos, presentaciones. Este actor maneja más privilegios que el estudiante.

Actores Pasivos:

- *Base de Datos*: es el componente dentro del sistema encargado de almacenar los datos obtenidos de la interacción entre los actores.

✓ **Identificación y Descripción de los Casos de Uso**

Se identifican los siguientes casos de uso para los siguientes actores:

- *Docente/Estudiante*: tiene 4 casos de uso en común:
 - Iniciar/cerrar sesión en el entorno virtual de aprendizaje (EVA).
 - Actualizar datos.
 - Administrar contenidos (visualizar recursos, cargar archivos).
 - Responder Mensajes/foros/comentarios.
- El **Docente** cuenta con un caso de uso adicional que difiere del estudiante:
 - Plantear tareas (tareas, foros, evaluaciones).

Para el caso de *iniciar/cerrar sesión*, el docente/estudiante tiene que ingresar su ID y password para ingresar al Entorno Virtual de Aprendizaje (EVA), y cuando termina su interacción con el entorno procede a cerrar sesión.

En el caso de actualizar datos, el docente/estudiante tiene la posibilidad de editar su información disponible en el EVA.

Administrar contenidos, hace referencia a que el docente / estudiante puede visualizar y seleccionar el componente académico para tener acceso a los recursos/archivos disponibles, selecciona el recurso que desea visualizar/ descargar, en el caso de cargar un archivo se selecciona y se lo carga en el sistema.

Responder mensajes/foros/comentarios, el docente/estudiante puede responder mensajes, foros o comentarios, según su necesidad, ingresa su respuesta y el sistema la registra. En el caso del estudiante puede enviar mensajes al docente en caso de tener alguna inquietud para obtener recursos adicionales.

Plantear tareas, el docente puede agregar, modificar o eliminar los recursos educativos plantear foros y formular evaluaciones, para medir el nivel de conocimiento del estudiante y ofrecerle una guía o un fundamento teórico para una mejor comprensión del componente.

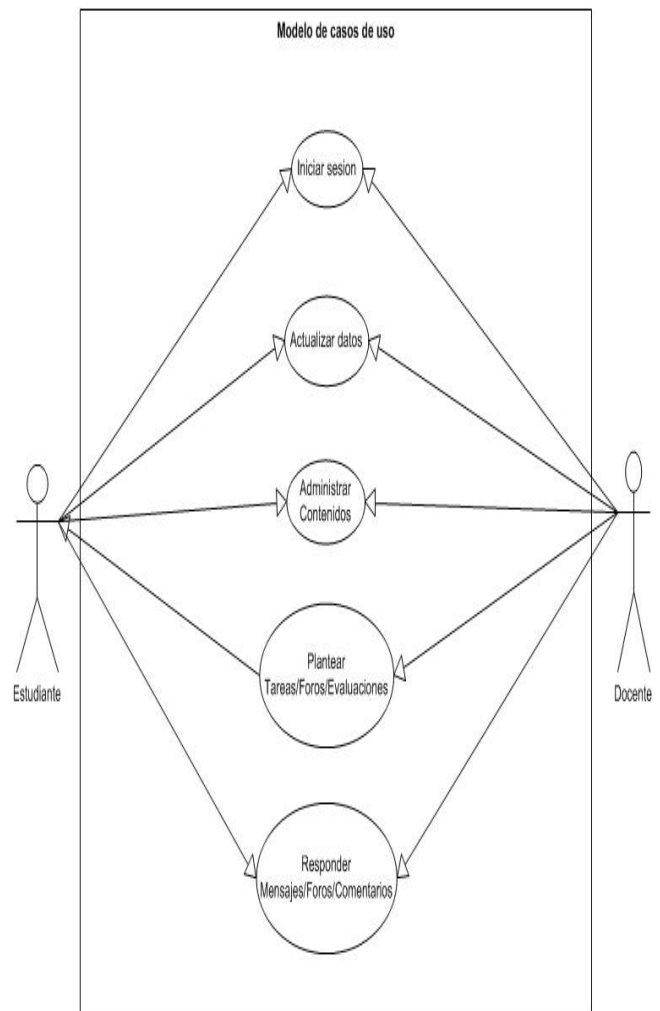


FIGURA 1. Casos de Uso para los Actores Activos

2.2 ANÁLISIS

Consiste en realizar un análisis de los componentes que formaran parte de la arquitectura, a partir de estos se puede realizar el diseño tanto físico y lógico, para la extracción de emociones en texto.

✓ **Modelo de Agente**

Especifica las características de un agente, se orienta esencialmente a los servicios.

■ **Identificación y descripción de los Agentes**

Para poder determinar los agentes del sistema se consideró la diferencia entre las tareas y los roles de los actores.

- **Agentes Humanos**, son los actores activos identificados en la fase de conceptualización, y son: Docente y Estudiante
- **Agentes Software**, una vez analizadas las tareas, se determinó los siguientes agentes: Extracción, Segmentación, Lematización, Ontología, Polaridad, Organizador.

Como ejemplo tomamos la conversación “Clasificación de Palabras”, cuyo objetivo es determinar la polaridad del texto y envía esta información al docente, el agente que interviene en esta conversación es el de Polaridad que es activado por el agente Organizador. El objetivo del agente Polaridad es determinar si el texto ingresado es positivo o negativo y el tipo de emoción que presenta, mediante la utilización de un diccionario afectivo, raíces, intensificadores, verbos, negaciones, lo almacena en una BD_Emociones y en el Agente Ontología, para que pueda cumplir con su tarea es necesario que el agente Organizador le envíe la información que recolecta del agente PLN.

▪ **Descripción de las Intervenciones**

La descripción de intervenciones tiene como objetivo determinar el intercambio de mensajes entre los agentes. Para realizar la descripción de las intervenciones gráficamente utilizamos el diagrama de secuencia de mensajes.

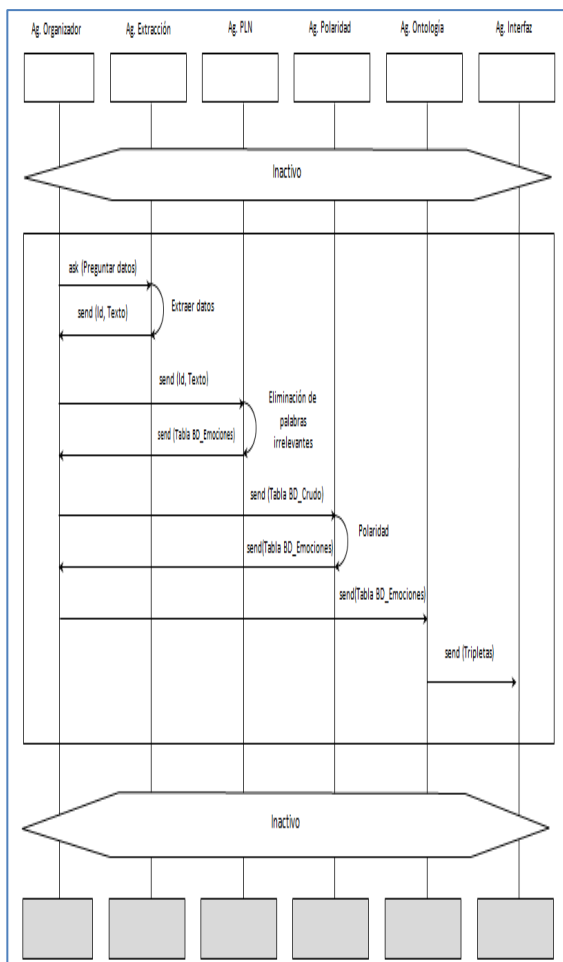


FIGURA 4. Diagrama de secuencias para las interacciones de la conversación

▪ **Canales básicos de Comunicación**

Una vez definidas todas las conversaciones, podemos determinar que mensajes son válidos entre los agentes,

luego los representamos mediante un diagrama resumen como se muestra en la FIGURA 5.

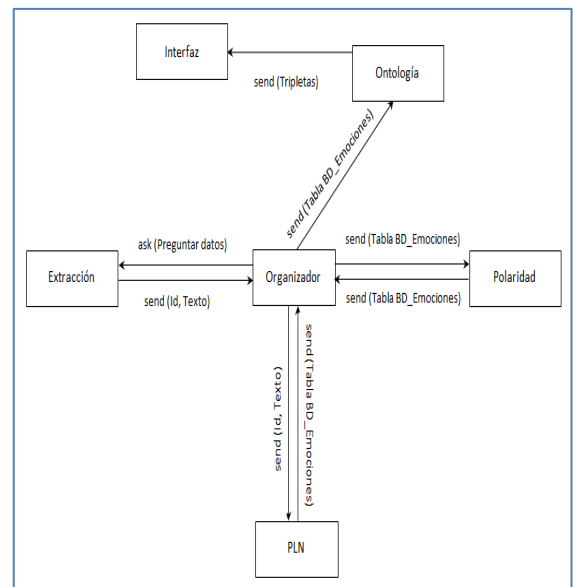


FIGURA 5. Canales básicos de comunicación y flujo de conversaciones entre agentes

✓ **Modelo de Comunicación**

Modela o describe las interacciones hombre-máquina, se descompone el diálogo entre el programa y el usuario utilizando un diagrama de secuencia de mensajes. Se considera los factores humanos en las interacciones.

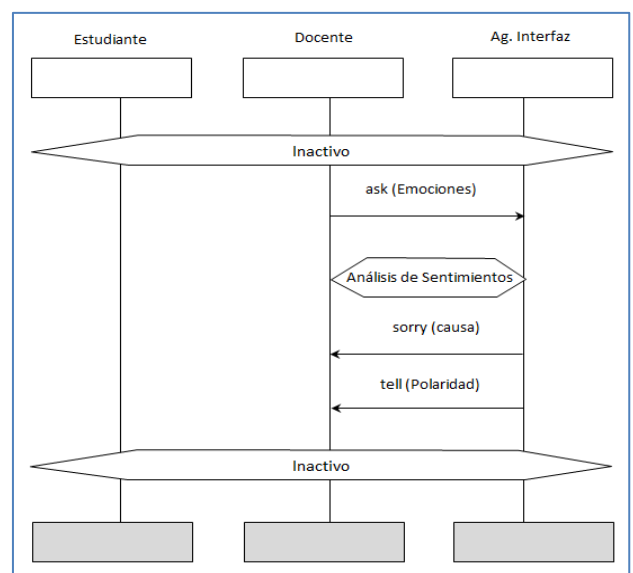


FIGURA 6. Interacciones genéricas de la conversación

✓ **Modelo de la Organización**

El objetivo es modelar las relaciones “estáticas” entre los agentes del sistema.

▪ **Identificación de las Relaciones de Herencia**

El objetivo es modelar las relaciones “estáticas” entre los agentes que componen el sistema.

El agente Organizador es la clase principal del cual el resto de agentes hereda atributos principales para su funcionamiento.

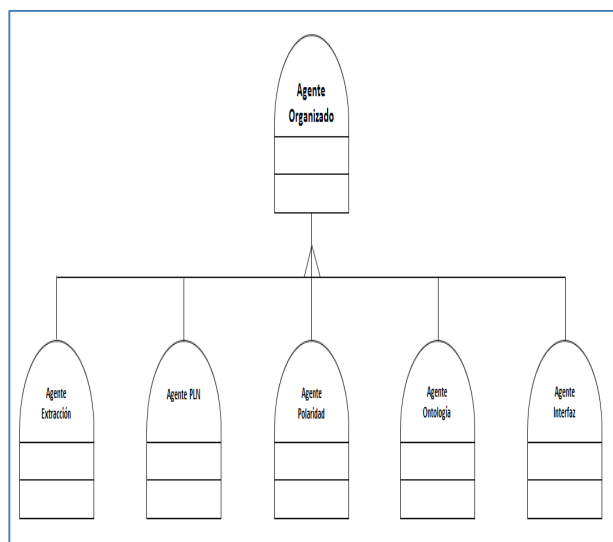


FIGURA 7. Jerarquía de Clases

▪ **Identificación de los Objetos del Entorno**

No hay un entorno muy definido dado que los agentes no son físicos, y no fue necesario definir sensores ni actuadores en las plantillas textuales de los agentes (Iglesias, 1998).

3. ARQUITECTURA PROPUESTA

Una vez obtenidos todos los requerimientos, pasamos a la fase de diseño y se obtiene como resultado la Arquitectura Multiagente y su implementación en JADE.

3.1 Esquema del Agente detección de Emociones en Texto

En la Figura 8 se muestra el esquema general del Agente Simple de detección de emoción en texto, en el mismo se puede observar como los agentes que componen la arquitectura reciben las percepciones de su entorno (EVA), mediante un analizador de patrones captan las señales (sensores) y son enviadas a los agentes se encargan de realizar el procesamiento de la información recibida y finalmente presentan el tipo de emoción encontrada. La Arquitectura recibe como parámetro de entrada texto obtenido de la mensajería, red social y foros, luego los procesa mediante el uso de un stopword list, diccionario afectivo, intensificadores, verbos, negaciones.

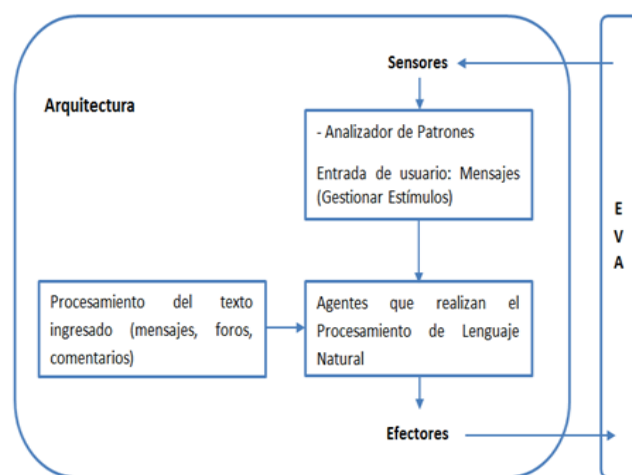


FIGURA 8. Esquema de Agente Simple detección de emociones en texto

3.2 Componentes

Dentro de los componentes que constituyen la arquitectura multiagente tenemos:

- **Estudiante:** Es la persona que interactúa con el entorno virtual de aprendizaje (EVA), de su interacción se obtiene los mensajes, foros, comentarios, y mensajes de la red social, para su posterior procesamiento.
- **Docente:** Es la persona que interactúa con la aplicación, para realizar el análisis de las emociones de los estudiantes.
- **Almacenamiento de resultados:** Es el almacenamiento de los resultados del análisis de texto, en BD_Emociones y en la Ontología, lo que almacenan es el tipo de emoción y el valor cuantitativo.
- **Obtención y Almacenamiento de datos:** Mediante el Agente Extracción, se obtiene información del entorno (mensajes, foros, comentarios), y se almacena en la base de datos BD_Emociones, luego es enviada al agente Organizador quien se encarga de enviar todo lo extraído al resto de agentes y coordina su funcionamiento.
- **Procesamiento de Texto:** Aquí se describe los agentes que se implementaron para el procesamiento de texto. El agente que interviene en este componente es el de PLN y los resultados que proporcionan este agente son almacenados en la BD_Emociones, y el Agente Organizador se encarga de enviarlo secuencialmente al agente Polaridad para que pueda procesar la información y presente el tipo de emoción encontrada.

3.3 Arquitectura Propuesta

Basándonos en los requisitos obtenidos en los modelos de la metodología MAS_COMMONKADS se propone la siguiente Arquitecturas basadas en el estándar FIPA:

✓ Arquitectura detección de Emociones en Texto

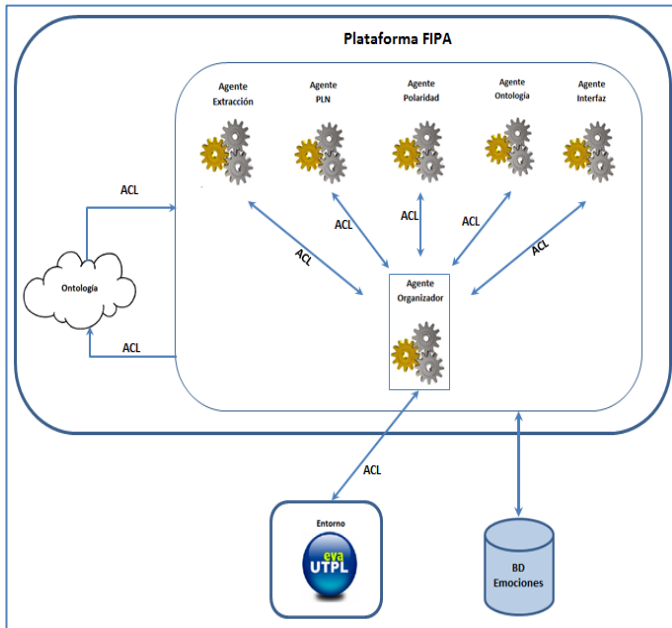


FIGURA 9.14 Arquitectura propuesta detección de Emociones en Texto

3.3.1 Comunicación entre Agentes

El lenguaje de comunicación entre agentes es un tema importante, ya que nos permite compartir y enviar mensajes entre los agentes, pero es necesario que estos mensajes tengan significado (contenido semántico), por esta se considera la utilización de ACL-FIPA.

ACL-FIPA es un lenguaje estandarizado para la comunicación de agentes, su principal objetivo es darle sentido semántico a los mensajes intercambiados entre los agentes. Para conseguir esto se basaron principalmente en los actos comunicativos para solicitar una acción a un agente. Para realizar una acción el agente "A" envía un mensaje *request* al agente "B"; el agente que recibe el mensaje puede aceptar (*accept*) o rechazar (*refuse*) la acción. En caso de que el agente "B" acepte la petición tiene que notificar e indicar al agente "A" cuando la realización de la acción finalice a través del mensaje *agree*. En caso de que la realización de la acción falle se notificara al agente "A" mediante el envío de un mensaje *failure* (Pinzón, 2010).

3.4 Desarrollo de la Arquitectura detección de emociones en Texto

Lenguaje de Programación, hemos seleccionado JAVA, básicamente se utilizó del framework JADE desarrollado para la creación de sistemas multiagente.

Implementación de los Agentes

Con la finalidad de ilustrar el procesamiento de datos que realiza cada agente se utilizó la técnica de diagramas de flujo para la representación gráfica del proceso, además se presenta parte del pseudocódigo desarrollado para la implementación de los agentes

Diagramas de Agentes

✓ Agente Polaridad

Determina la polaridad del texto y el tipo de emoción, toma como parámetro de entrada el resultado del agente PLN, utiliza intensificadores, negaciones, verbos, palabras raíces y un diccionario afectivo, para poder obtener la polaridad del texto.

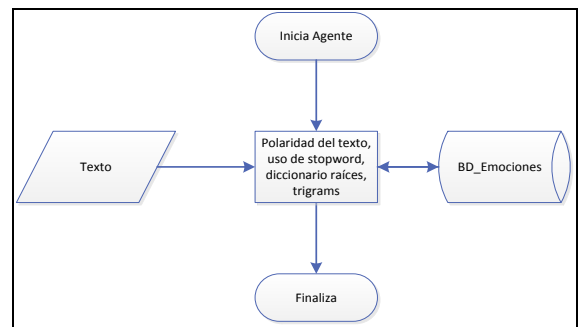


FIGURA 150. Diagrama de flujo Agente Polaridad

Desarrollo de la Arquitectura

✓ Herramienta gráfica

A continuación se presenta la herramienta gráfica RMA (Remote Monitoring Agent) propia de JADE, en la cual podemos inicializar los agentes. Inicializamos los agentes que intervienen en la arquitectura, Ontología, Polaridad e Interfaz.

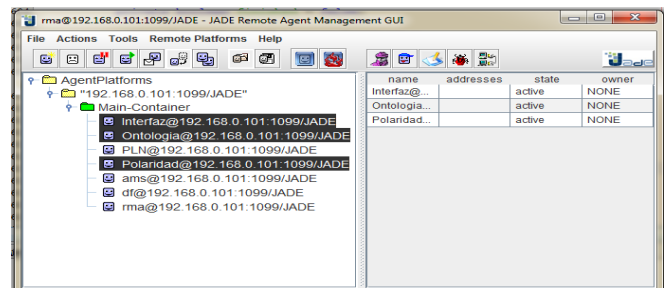


FIGURA 11.16 Interfaz gráfica del RMA

El sistema multiagente nos permite realizar el análisis de texto, obtiene su polaridad y el tipo de emoción detectada, procesa varios mensajes a la vez y se presenta el resultado por cada mensaje analizado y lo almacena en archivo .txt.

En el botón Cargar, seleccionamos el archivo que deseamos procesar, luego damos clic en el botón Realizar- Proceso, y podemos visualizar en la interfaz gráfica el resultado y también en un archivo .txt. El tiempo de ejecución y muestra de resultados dependerá del tamaño del corpus con

el que trabajemos. Con fines educativos se tomó varios mensajes de la Red Social del Eva y se obtuvo los siguientes resultados:

Resultados obtenidos de los mensajes:

- Los mensajes utilizados para las pruebas se encuentra en los Anexos1-A.

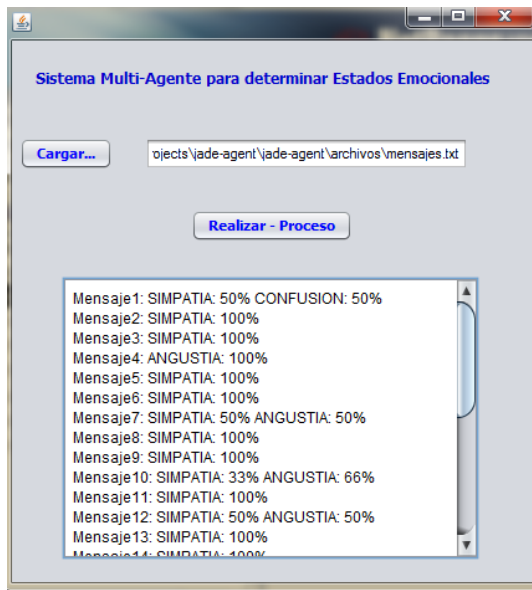


FIGURA 117 Resultados del Sistema

Como podemos observar en la FIGURA 12 el resultado obtenido se muestra por cada mensaje analizado, se realizó varias pruebas y se tiene como resultado global que las emociones simpatía, confusión, angustia y frustración, están presentes o influyen con mayor frecuencia en el aprendizaje de los estudiantes, lo cual trae como consecuencia el bajo rendimiento de los mismos, por esta razón es conveniente que el docente conozca esta información para que puede brindar una retroalimentación a los estudiantes en los componentes académicos.

Además, cabe recalcar que el resultado obtenido del procesamiento de texto por parte de nuestro sistema multiagente fue comparado con un análisis previo que se realizó mediante la utilización de escenarios, de los cuales se obtuvo varias características que nos ayudan a determinar la polaridad de las emociones y el tipo de emoción, dichas características fueron plasmados en el sistema multiagente.

CONCLUSIONES

Mediante el desarrollo e implementación de la arquitectura multiagente, se puede concluir lo siguiente:

- ✓ El Análisis de Sentimientos intenta clasificar los documentos según su polaridad, mediante la utilización de PLN, para determinar ciertos patrones que nos permitan conocer que emociones afectan o están presentes en el aprendizaje.
- ✓ La metodología MAS_CommonKADS nos permite el desarrollo de sistemas multiagente mediante la utilización de modelos, para obtener todos los

requerimientos necesarios para la implementación de la Arquitectura, cabe recalcar que esta metodología nos ofrece un marco de especificaciones independiente de la implementación.

- ✓ JADE es un marco de desarrollo para la creación de sistemas multiagente, nos brinda interoperabilidad, portabilidad y está basado en los estándares FIPA para definir protocolos de comunicación, además nos permite trabajar con ontologías y permite el trabajo peer-to-peer.
- ✓ Es conveniente la utilización de un StopWord, para la eliminación de palabras irrelevantes que contienen los mensajes, ya que no representan ningún tipo de emoción.
- ✓ El planteamiento de escenarios es un punto importante para el desarrollo de este proyecto, porque nos ayuda a conocer e interpretar las alteraciones lingüísticas que presentan los mensajes, de esta forma se pueden obtener ciertas palabras clave que nos ayudan a determinar los tipos de emociones presentes en los mismos.
- ✓ La aplicación de la metodología MAS_CommonKADS nos permite obtener todos los requerimientos necesarios para implementar la Arquitectura que se plantea.
- ✓ Las emociones detectadas mediante el análisis de los mensajes con mayor ocurrencia son: Confusión y Angustia. Estas emociones nos indican que los estudiantes necesitan una retroalimentación en algunos temas de estudio.

RECOMENDACIONES Y TRABAJOS FUTUROS

- ✓ Es necesario seguir una metodología de desarrollo de sistemas multiagente, ya que en su desarrollo se establece todos los requerimientos necesarios para su posterior implementación.
- ✓ Para comprender mejor en análisis textual, es conveniente conocer algunos temas concernientes a ciertas alteraciones de las formas lingüísticas, que se encuentran presentes en los mensajes.
- ✓ Plantear escenarios es importante para poder obtener ciertas características que influyen en los estados emocionales de los estudiantes.
- ✓ Para la implementación de la arquitectura propuesta, es necesario analizar todas las herramientas disponibles y elegir la que mejor se adapte a nuestros requerimientos.

Al ser un tema de actualidad tiene mucho potencial, se puede desarrollar aplicaciones móviles para realizar análisis de sentimientos, y según el estado de ánimo del usuario, cambiar el tono de llamada o mensaje de su teléfono móvil.

Aplicación educativa, el uso de las redes sociales como punto de partida para el análisis psicológico y emocional de los estudiantes.

Análisis del rendimiento educativo, según el estado emocional del estudiante.

REFERENCIAS BIBLIOGRÁFICAS

Akbiyik, C. (2010). ¿Puede la informática afectiva llevar a un uso más efectivo de las Tecnologías de la Información y de la Comunicación (TIC) en la Educación? *Revista de Educación*(352), 190.

Baggia, P., Burkhardt, F., Pelachaud, C., Peter, C., & Zovato, E. (2012). Emotion Markup Language (EmotionML) 1.0. Obtenido de W3C incubator group report, World Wide Web Consortium: <http://www.w3.org/TR/emotionml/>

Barrett, L. F., & Russell, J. A. (1999). The Structure of Current Affect: Controversies and Emerging Consensus. Obtenido de American Psychological Society: <http://www.affective-science.org/pubs/1999/FBRussell99.pdf>

Bellifemine, F., Caire, G., & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Wiley.

Blanco, A., & Carracedo, A. (2009). Inteligencia Emocional. Obtenido de Universia: <http://dspace.universia.net/bitstream/2024/195/1/Trabajo+imprimible.pdf>

Borges, F. (2005). La frustración del estudiante en línea. Causas y acciones preventivas. Obtenido de DIGITHUM: <http://www.uoc.edu/digithum/7/dt/esp/borges.pdf>

Bratman, M. (1987). *Intentions, Plans, and Practical Reason*. Harvard University Press.

Breiman, L. (1994). Obtenido de <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=506D5DF160BDEDFE92B85A1261E1549F?doi=10.1.1.32.9399&rep=rep1&type=pdf>

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Boca Raton, Florida: CHAPMAN & HALL.

Cabello, E., Conde, C., Martín, I., & Serrano, Á. (2006). Técnicas de Reconocimiento Automático de Emociones. *Revista Electrónica Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*. Universidad de Salamanca, 107-127.

Carrillo, J. (2011). Un Modelo Lingüístico-Semántico Basado en Emociones para la Clasificación de Textos según su Polaridad e Intensidad. Obtenido de Natural Language Processing and Information Retrieval Group at UNED: http://nlp.uned.es/~jcalbornoz/papers/PhD_Thesis_2011.pdf

Chuang, Z.-J., & Wu, C.-H. (2004). Multi-Modal Emotion Recognition from Speech and Text. Obtenido de The Association for Computational Linguistics:

<http://www.aclweb.org/anthology-new/O/O04/O04-3004.pdf>

Corchado, J. M. (2005). Universidad de Salamanca: BISITE (Bioinformática, Sistemas Inteligentes, Tecnología Educativa). Obtenido de Modelos y Arquitecturas de Agentes: [http://bisite.usal.es/archivos/c1%20\(1\).pdf](http://bisite.usal.es/archivos/c1%20(1).pdf)

Cortijo, F. (2000). Técnicas supervisadas II: Aproximación no paramétrica. Obtenido de Instituto de Ingeniería Eléctrica. Universidad de Granada.: http://iie.fing.edu.uy/ense/assign/recpat/material/tema3_00-01/node25.html

DeLoach, S. (2011). Analysis and Design using MaSE and agentTool. Obtenido de CiteSeerX: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=E1D62BF60F8D3FCD31655B74C51F55D?doi=10.1.1.155.3813&rep=rep1&type=pdf>

Fakhraie, N. (2012). What's in a Note? Sentiment Analysis in Online Educational Forums. Obtenido de https://tspace.library.utoronto.ca/bitstream/1807/30086/1/Fakhraie_Seyedeh_Najmeh_201111_MA_thesis.pdf

Fausett, L. (1994). *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications*. New Jersey, USA: Prentice-Hall.

Ferber, J. (1999). *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*. Harlow: Addison Wesley Longman.

Freund, Y., & Schapire, R. (1996). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Obtenido de Journal of Computer and System Sciences: http://www.face-rec.org/algorithms/Boosting-Ensemble/decision-theoretic_generalization.pdf

Goleman, D. (1996). *Inteligencia Emocional*. Barcelona: Kairós.

Greenleaf, R. K. (2003). Motion and Emotion. Obtenido de Principal Leadership: <http://www.principals.org/portals/0/content/46875.pdf>

Hernandez, J. (2003). Proyecto DAMMAD, Diseño y Aplicación de Modelos Multiagente para Ayuda a la Decisión, TIC2000-1370-C04-02. Obtenido de Grupo de Investigación de Sistemas Inteligentes. Universidad Politécnica de Madrid: <http://platon.escet.urjc.es/grupo/proyectos/dammad/propio/Documentos/Informes/ArquitecturasAgente.doc>

Iglesias, C. (1998). Obtenido de <http://www.upv.es/sma/teoria/agentes/tesiscif.pdf>

Iglesias, C. (1998). Definición de una Metodología para el Desarrollo de Sistemas Multiagente (Tesis Doctoral). Obtenido de Universidad Politécnica de Madrid: <http://www.upv.es/sma/teoria/agentes/tesiscif.pdf>

- Iglesias, C., Garijo, M., & González, J. (1999).
- Imbert, R. (2005). Una Arquitectura Cognitiva Multinivel para Agentes con Comportamiento Influído por Características Individuales y Emociones, Propias y de Otros Agentes. Obtenido de Archivo Digital UPM-Biblioteca Universitaria-Universidad Politécnica de Madrid: http://oa.upm.es/477/1/Ricardo_Imbert.pdf
- Kogan, P., & Roger, S. (2011). Universidad Nacional de La Plata (UNLP). Grupo de Investigación en Lenguajes e Inteligencia Artificial. Obtenido de Análisis de opinión como un sistema multiagente distribuido: http://sedici.unlp.edu.ar/bitstream/handle/10915/19999/Documento_completo.pdf?sequence=1
- Kort, B., Reilly, R., & Picard, R. W. (2001). An Affective Model of Interplay Between Emotions and Learning: Reengineering Educational Pedagogy - Building a Learning Companion. Obtenido de <http://affect.media.mit.edu/projectpages/lc/icalt.pdf>
- Leshed, G., & Kaye, J. (2006). Understanding How Bloggers Feel: Recognizing Affect in Blog Posts. Obtenido de <http://alumni.media.mit.edu/~jofish/writing/recognizing-bloggers-affect.pdf>
- Liu, H., Lieberman, H., & Selker, T. (2003). A Model of Textual Affect Sensing using Real-World Knowledge. Obtenido de http://pop.larifari.org/_/writing/IUI2003-AffectSensing.pdf
- Maes, P. (1990). Designing Autonomous Agents (Vol. 6). Amsterdam, Netherlands: Elsevier Science Publishers B. V.
- Martín, I., Serrano, Á., Conde, C., & Cabello, E. (2006). Técnicas de Reconocimiento Automático de Emociones. Revista Electrónica Teoría de la Educación. Educación y Cultura en la Sociedad de la Información., 107-127.
- Martín, M. (2012). Sistema de Clasificación Automática de Críticas de Cine. Obtenido de Proyecto de Fin de Carrera: http://e-archivo.uc3m.es/bitstream/10016/5846/1/PFC_Miriam_Martin_Garcia.pdf
- Martínez, E., Martín, M., Perea, J., & Ureña, L. (2011). Técnicas de clasificación de opiniones aplicadas a un corpus en español. Obtenido de RUA. Repositorio Institucional de la Universidad de Alicante: http://rua.ua.es/dspace/bitstream/10045/18524/1/PLN_47_17.pdf
- McQuiggan, S. W., Lee, S., & Lester, J. C. (2007). Early Prediction of Student Frustration. Obtenido de Department of Computer Science, North Carolina State University, Raleigh, NC 27695: <http://www.intellimedia.ncsu.edu/papers/frustration-acii-2007.pdf>
- Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.
- Nwana, H., Ndumu, D., Lee, L., & Collis, J. (1999). ZEUS: A Toolkit for Building Distributed Multi-Agent Systems. Obtenido de CiteSeerX: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.277&rep=rep1&type=pdf>
- Oatley, K. (1992). Best Laid Schemes: The Psychology of Emotions. Cambridge University: Maison des Sciences de l'Home.
- Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. Obtenido de Cornell University-Department of Computer Science: <http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf>
- Pekrun, R. (2006). The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice. Obtenido de http://www.scipie.net/docs/2011/area2/Pekrun_EPR_2006.pdf
- Perozo, N. (2011). Modelado Multiagente para Sistemas Emergentes y Auto-Organizados (Tesis doctoral). Obtenido de Universidad de los Andes- Facultad de Ingeniería Doctorado en Ciencias Aplicadas: <http://www.ing.ula.ve/~oteran/tesisPerozo.pdf>
- Rebollo, M. Á., García, R., Barragán, R., Buzón, O., & Vega, L. (2008). Las Emociones en el Aprendizaje Online. RELIEVE, v.14, n. 1, p. 2.
- Rizo, R., Llorens, F., & Pujol, M. (2003). Universidad de Alicante, Dpto. de Ciencia de la Computación e Inteligencia Artificial. Obtenido de Arquitecturas y Comunicación entre Agentes: <http://www.dccia.ua.es/dccia/inf/asignaturas/AI/docs/ACA.pdf>
- Schröder, M., Baggia, P., Burkhardt, F., Pelachaud, C., Peter, C., & Zovato, E. (2011). EmotionML- an upcoming standard for representing emotions and related states. Obtenido de Deutsches Forschungszentrum für Künstliche Intelligenz: http://www.dfki.de/~schroed/articles/schroeder_etal2011b.pdf
- Sebe, N., Sun, Y., Bakker, E., Lew, M., Cohen, I., & Huang, T. (2004). Towards Authentic Emotion Recognition. Obtenido de CiteSeerX: <http://citeseerx.ist.psu.edu/viewdoc/download?jsessionid=E91EA6EAA9965AEE62201764D7741230?doi=10.1.1.107.4058&rep=rep1&type=pdf>
- Seligman, M., Walker, E., & Rosenhan, D. (2001). Abnormal psychology (4th ed.). New York: W. W. Norton & Company, Inc.
- Télez, A. (2005). Extracción de Información con Algoritmos de Clasificación. Obtenido de inaoe (Instituto

Nacional de Astrofísica, Óptica y Electrónica):
<http://ccc.inaoep.mx/~mmontesg/tesis%20estudiantes/TesisMaestria-AlbertoTellez.pdf>

Turney, P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviewa. Philadelphia: 40th Anual Meeting of the Association for Computational Linguistics (ACL). Obtenido de <http://acl.lidc.upenn.edu/P/P02/P02-1053.pdf>

Turney, P., & Littman, M. (2003). Measuring Praise and Criticism: Inference of Semantic Orientation from Association. Obtenido de <http://cogprints.org/3164/1/turney-littman-acm.pdf>

Vapnik, V. (1999). An Overview of Statistical Learning Theory. Obtenido de Massachusetts Institute of Technology: http://web.mit.edu/6.962/www/www_spring_2001/emin/slt.pdf

Wesson, R., & Hayes-Roth, F. (1980). RAND CORPORATION. Obtenido de Network Structures for Distributed Situation Assessment: <http://www.rand.org/content/dam/rand/pubs/reports/2006/R2560.pdf>

Wooldridge, M., Jennings, N., & Kinny, D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. Obtenido de University of Southampton: <http://eprints.soton.ac.uk/253748/1/jaamas2000.pdf>