



DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**"SISTEMA MULTI-AGENTE DE RECONOCIMIENTO DE FRASES  
EN LSM UTILIZANDO CBR."**

**TESIS**

PRESENTADA POR:

**ING. CÉSAR RENÉ MARTÍNEZ AGUIRRE**

COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

**Directora de Tesis: M.C. ANA LUISA MILLÁN CASTRO**

**HERMOSILLO, SONORA, MÉXICO**

**03 DE SEPTIEMBRE 2020**





"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

SECCIÓN: DIV. EST. POS. E INV.  
No. OFICIO: DEPI/064/20  
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN  
DE TESIS.

22 de Junio de 2020

**C. CÉSAR RENÉ MARTÍNEZ AGUIRRE,  
PRESENTE.**

Por este conducto, y en virtud de haber concluido la revisión del trabajo de tesis que lleva por nombre "SISTEMA MULTI-AGENTE DE RECONOCIMIENTO DE FRASES EN LSM UTILIZANDO CBR", que presenta para el examen de grado de la MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN, y habiéndola encontrado satisfactoria, nos permitimos comunicarle que se autoriza la impresión del mismo a efecto de que proceda el trámite de obtención de grado.

Deseándole éxito en su vida profesional, quedo de usted.

ATENTAMENTE

M.C. ANA LUÍSA MILLÁN CASTRO  
DIRECTORA

M.C. CÉSAR ENRIQUE ROSE GÓMEZ  
SECRETARIO

A. Mancinas

DR. ABELARDO MANCINAS GONZÁLEZ  
VOCAL

M.C. ROSA IRENE SÁNCHEZ FERMÍN  
JEFA DE LA DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



S.E.P.

RISF/fjcg\*

INSTITUTO TECNOLÓGICO  
DE HERMOSILLO  
DIVISIÓN DE ESTUDIOS  
DE POSGRADO



## **Resumen**

Es sabido que, a las personas con capacidades diferentes, se les dificulta relacionarse en su vida cotidiana.

Una de las discapacidades más comunes y que también afecta mucho en varios factores de la vida de quienes la sufren, es la sordera. Debido a esta condición, los Sordos han tenido que adaptarse a su entorno mediante una comunicación diferente a como sería habitualmente.

En esta tesis se propone el desarrollo de un Sistema Multi-Agente (SMA) capaz de reconocer frases que son convertidas a Lengua de Señas Mexicana (LSM) a través de un avatar en un dispositivo móvil. Para esto, el SMA emplea la técnica de Razonamiento Basado en Casos (CBR por sus siglas en inglés) con el fin de proveer al SMA de la capacidad para comunicarse ante la presencia de nuevas frases.

## Índice General

Introducción.....	2
Capítulo 1: Planteamiento del Problema .....	5
1.1. Antecedentes.....	5
1.2. Definición del Problema.....	7
1.3. Preguntas de Investigación .....	7
1.4. Objetivos .....	7
1.4.1. Objetivo General .....	8
1.4.2. Objetivos Específicos:.....	8
1.5. Justificación.....	8
1.6. Alcances y Delimitaciones.....	9
Capítulo 2: Estado del Arte.....	10
2.1. Lengua de Señas Mexicana .....	11
2.2. Razonamiento Basado en Casos .....	12
2.3. Agentes Inteligentes .....	17
2.3.1. Sistemas Multi-Agentes .....	23
2.3.2. Arquitecturas Multi-Agentes .....	24
2.3.2.1. Arquitecturas Reactivas .....	24
2.3.2.2. Arquitecturas Híbridas .....	26
2.4. Animaciones Mediante Avatar .....	26
2.4.1. Opciones de Diseño para un Avatar .....	27
2.5. Trabajos Relacionados.....	28
2.5.1. Chinese Sign Language Animation System On Mobile Devices .....	28
2.5.2. IBM System Is a Virtual Sign-Laguage Interpreter .....	28
Capítulo 3: Análisis y Diseño.....	30
3.1 Diagrama de Contexto.....	30
3.1.1 Diagrama de Contexto de Nivel 0.....	30
3.1.2 Diagrama de Contexto de Nivel 1 .....	31
3.2. Arquitectura del Sistema Propuesto .....	32
3.3. Diagramas .....	33
3.3.1. Casos de Uso .....	33
3.3.2. Modelo de Objetivos .....	36

3.3.3. Modelo de Organización .....	39
3.3.4. Modelo de Agentes .....	40
3.3.5 Modelo de Interacciones .....	42
Capítulo 4: Implementación.....	44
4.1 Modelo de Interacciones .....	44
4.2 Creando la Base de Creencias .....	44
4.2.1 Estableciendo la Conexión.....	45
4.3. Manejo de Frases .....	46
4.3.1. Separación de Palabras .....	47
4.3.2. Primera Tabla Palabras .....	49
4.3.3. Segunda Tabla Palabras.....	50
4.3.4. Primera y Segunda Tabla Frases.....	52
4.4. Agentes Coordinador e Interfaz .....	54
4.4.1. Creando al Agente Coordinador y al Agente Interfaz.....	55
4.4.2. Grafo de flujo del Sistema .....	59
Capítulo 5: Análisis de resultados .....	61
5.1. Pruebas del Sistema .....	61
5.1.1. Pruebas de Ruta Básica.....	62
5.1.2. Interpretación de los resultados .....	64
Capítulo 6: Conclusiones y Trabajo Futuro .....	78
6.1. Conclusiones.....	79
6.2. Trabajo Futuro.....	80
Referencias .....	82

## Índice Figuras

1.1. Personas con Discapacidad .....	3
2.1. Diagrama CBR .....	13
2.2. Diagrama Funcionamiento Agente .....	20
3.1. Diagrama de Contexto Nivel 0.....	30
3.2. Diagrama de Contexto Nivel 1.....	31
3.3. Arquitectura Propuesta.....	32
3.4. Casos de Uso.....	34
3.5. Diagrama Modelo de objetivos .....	36
3.6. Objetivo Obtener Frase .....	37
3.7. Objetivo Reconocer Frase.....	38
3.8. Objetivo Proyectar Frase.....	39
3.9. Diagrama Modelo Organización .....	40
3.10. Diagrama Modelo de Agentes .....	41
3.11. Diagrama de Tareas.....	42
3.12. Diagrama de Interacciones.....	42
4.1. Tablas de la Base de Creencias.....	45
4.2. Conexión a Base de Creencias .....	45
4.3. Librería de MySQL .....	46
4.4. Ingreso de Frase .....	47
4.5. Separación de Palabras .....	48
4.6. Comparación de Palabras Primera Tabla.....	50
4.7. Comparación de Palabras Segunda Tabla.....	51
4.8. Cracion de Tablas “frase” y “frase2” .....	52
4.9. Interfaz del IDE JADE .....	55
4.10. Clase del Agente Coordinador .....	56
4.11. Clase del Agente Interfaz .....	57
4.12. Propiedades del Proyecto .....	58
4.13. Librerías Ageregadas a las Propiedades del Proyecto.....	59
4.14. Grafo del Sistema.....	60
5.1. Casos de Prueba Obtenidos .....	63
5.2. Resultados de Ejecución del Sistema del Caso 1 .....	68

5.3. Resultados de Ejecución del Sistema del Caso 2 .....	68
5.4. Resultados de Ejecución del Sistema del Caso 3 .....	69
5.5. Resultados de Ejecución del Sistema del Caso 4 .....	70
5.6. Resultados de Ejecución del Sistema del Caso 5 .....	71
5.7. Resultados de Ejecución del Sistema del Caso 6 .....	71
5.8. Resultados de Ejecución del Sistema del Caso 7 .....	72
5.9. Resultados de Ejecución del Sistema del Caso 8 .....	73
5.10. Resultados de Ejecución del Sistema del Caso 9 .....	73
5.11. Resultados de Ejecución del Sistema del Caso 10 .....	74
5.12. Interacción Entre Agentes .....	76
5.13. Información del Mensaje Entre Agentes.....	77
5.14. Resultado Final al Recibir Frase .....	78

## Índice Tablas

2.2. Ejemplos de aspectos de un agente .....	20
2.3. Ejemplos de entornos y agentes .....	22
3.1. Casos de Uso Obtener Frase.....	34
3.2. Casos de Uso Procesar Frase .....	34
3.3. Casos de Uso Enviar Frase.....	35
4.1. Descripción de estados de Sistema .....	60
5.1. Interpretación de Rutas de Prueba.....	64

## **Introducción**

Las personas Sordas, son aquellas que tienen dificultad o la imposibilidad de usar el sentido del oído debido a una pérdida de la capacidad auditiva, la cual se denomina como hipoacusia. La hipoacusia es uno de los síntomas que pueden estar presentes en la enfermedad otoneurología. La pérdida de audición constituye un motivo de consulta muy frecuente en atención primaria y sobre todo en las consultas de atención especializada de Otorrinolaringología.

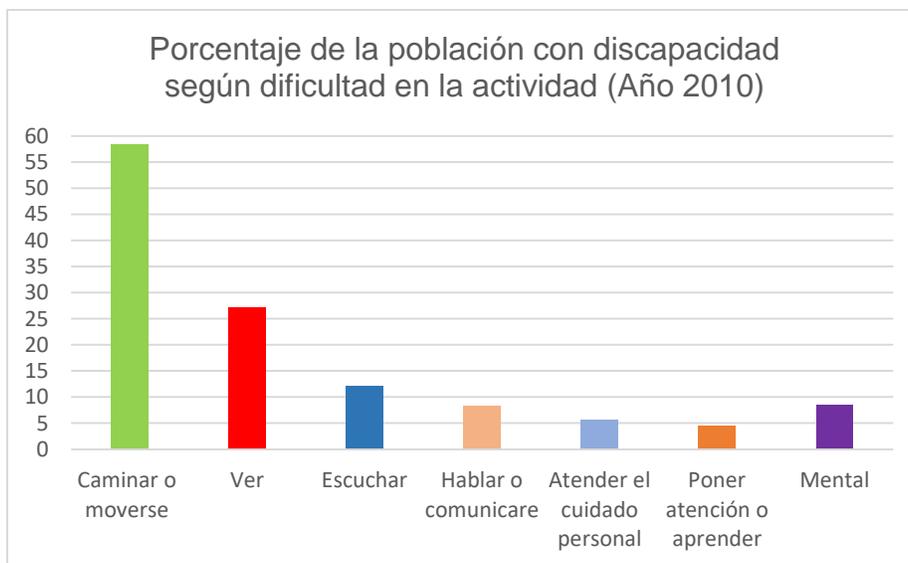
La hipoacusia puede ser motivo de varias patologías, desde una condición fácilmente tratable, hasta un proceso sistemático grave.

Existen varios grados dependiendo de la intensidad de esta condición, desde casos leves hasta casos muy severos, llegando a ser invalidantes, afectando considerablemente en la forma en que la persona Sorda se relaciona en su entorno social, al encontrarse con una seria limitación en su capacidad de encontrar una vía de comunicación [1].

A nivel mundial la sordera está presente en millones de personas; sin embargo, dependiendo del país, se les otorgan ciertas facilidades y apoyos, así como a los ciegos indicaciones en braille y a las personas con discapacidad motriz, rampas o sujetadores. En México hay varios cientos de miles de personas Sordas, pero sin el apoyo necesario para poder sobrellevar esta discapacidad con mayor facilidad y tener una mejor calidad de vida.

Una de las comunidades con discapacidad más grande, es la de personas con problemas de audición, representando aproximadamente el 5% de la población mundial [2][3].

En la siguiente figura 1.1, se ilustra el porcentaje de la población con discapacidad en México, según la dificultad en la actividad [4](Año 2010).



*Figura 1.1 Personas con Discapacidad*

En México las personas con discapacidad auditiva corresponden del 10% al 12% del total de personas con alguna discapacidad, como se muestra en la figura 1.1. [4].

Se cuenta sólo con 40 traductores certificados en todo el país, siendo éstos en su mayoría familiares de algún familiar Sordo. De esta cifra de traductores certificados, 11 se encuentran en la capital del país y apenas ha aumentado desde el 2009, notándose así, la falta de apoyo que esta comunidad recibe por parte del gobierno, la cual presenta complicaciones tanto en ámbitos escolares, sociales, económicos y laborales, solamente por su dificultad para comunicarse. Siendo de estos aspectos el escolar uno de los más afectados y que impacta de manera significativa a los demás mencionados, puesto que, al momento de ingresar a niveles de educación superior, estas personas deben buscar la manera de adaptarse a un ambiente que no es adecuado para ellos, requiriendo en la mayoría de los casos contratar tutores sombra, que fungen como traductores en tiempo real. También se presenta el problema de que muchas de estas personas no se están certificadas, pudiendo así realizar interpretaciones erróneas, no sólo en situaciones cotidianas, sino en un diagnóstico médico o en un juicio con consecuencias graves [3].

Para ayudar a los Sordos con esta problemática, se propone un mecanismo de reconocimiento de palabras y/o frases transmitidas por un interlocutor, las cuales son capturadas por un dispositivo electrónico.

El proyecto consistirá en las siguientes fases:

- Investigación del estado del arte: Adentrarse en los ejes principales de la investigación; es decir, buscar documentos relacionados con los temas a tratar en esta tesis, adquirir información mediante entrevistas o con expertos en el tema y conocer la gramática en LSM.
- Análisis y diseño del sistema: Esta fase está basada en el desarrollo del SMA para modelar un módulo de razonamiento basado en CBR para reproducir frases existentes o en su defecto, crear nuevas frases a partir de casos previos.
- Implementación de la propuesta: Consistirá en realizar pruebas de casos para verificar la confiabilidad de los resultados esperados por el SMA.
- Resultados: En esta fase, se muestran los resultados obtenidos y las metas logradas el sistema.
- Conclusiones y trabajos futuros: Se verán los resultados finales obtenidos, las mejoras posibles al sistema que pudieran complementar el trabajo en futuros proyectos.

## **Capítulo 1: Planteamiento del Problema**

En este primer capítulo se presenta el origen de este proyecto, se describen los antecedentes del problema a resolver, así como la situación en la que viven las personas Sordas. Por otra parte, se presenta una justificación, en la cual se menciona el motivo por el cual se decidió realizar este proyecto, se muestran los objetivos y las preguntas de investigación, con las cuales se establecen las metas que se esperan cumplir y por último, el alcance de esta investigación.

### **1.1. Antecedentes**

La sordera se define como impedimento auditivo severo, que obstaculiza al individuo el procesamiento de información lingüística por vía auditiva, por lo tanto, la sordera puede ser vista como una condición que evita que una persona reciba sonido en todas o casi todas sus formas [5].

A nivel mundial, 466 millones de personas padecen pérdida de audición discapacitante, de las cuales 34 millones son niños. Se calcula que para el año 2050, casi mil millones de personas, sufrirá de esta condición [6].

La pérdida de la audición puede deberse a causas genéticas, complicaciones en el parto, algunas enfermedades infecciosas, infecciones crónicas del oído, el empleo de determinados fármacos, las exposiciones al ruido excesivo y el envejecimiento.

Los casos desatendidos de pérdida de audición representan un coste mundial anual de 750 000 millones de dólares internacionales [6]. Las intervenciones destinadas a prevenir, detectar y tratar la pérdida de audición no son caras y pueden resultar muy beneficiosas para los interesados.

Por pérdida de audición discapacitante se entiende una pérdida de audición superior a 40dB en el oído con mejor audición para adultos, y superior a 30dB en el oído con mejor audición en los niños. La mayoría de las personas con pérdida de audición severa vive en países de ingresos bajos y medianos [6].

La situación de las personas que padecen pérdida de audición se ve mejorada gracias a la detección temprana, al uso de audífonos, implantes y otros dispositivos de ayuda, así como el empleo de subtítulos, el aprendizaje del lenguaje de señas y otras medidas de apoyo educativo y social.

En nuestro país se tiene conocimiento de 5.7 millones de personas con alguna discapacidad, de los cuales casi 700 mil son Sordos (12% de la muestra total), de acuerdo con cifras del censo de población y vivienda, realizado por el Instituto Nacional de Estadística y Geografía [7].

Debido a la necesidad de mejorar la calidad de vida de los Sordos, se han presentado y desarrollado diferentes proyectos e ideas, como el uso de aparatos electrónicos, implantes, etc., como se mencionó anteriormente, el avance en la tecnología ha propiciado el desarrollo de nuevas propuestas, por ejemplo el prototipo de un guante, presentado en la tesis “Diseño y construcción de un guante prototipo electrónico capaz de traducir el lenguaje de señas de una persona Sordomuda al lenguaje de letras”, que por medio de sensores y micro controladores es capaz de traducir a letras la lengua de señas, según el movimiento captado en el guante por medio de los sensores [8], el funcionamiento de este guante es muy similar a lo que se propone con un avatar, y aunque la idea de traducir de lengua de señas al español igual es útil para las personas que no comprenden ese lenguaje, este prototipo se queda limitado, al solo contemplar las manos, cuando en la lengua de señas se utiliza también el torso completo, gestos de la cara, movimientos de los brazos, incluso movimientos con la boca, etc.

Otro trabajo similar es el titulado “Sistema de reconocimiento de voz y traducción a lengua de señas mediante un avatar”, el cual hace uso de animaciones tridimensionales, para la traducción de frases desde lenguaje natural a lenguaje de señas mexicano, donde la frase es detectada por medio de reconocimiento de voz, procesada y estructurada según la gramática del LSM y mediante un algoritmo, la frase es convertida en señas mediante animaciones previamente formadas, las cuales se visualizan a través de una aplicación móvil [9].

## **1.2. Definición del Problema**

La discapacidad auditiva limita en muchos sentidos la calidad de vida de una persona, haciendo que la misma presente problemas para comunicarse y en su desenvolvimiento social [2]. Estas personas cuentan con un sin fin de dificultades durante toda su vida por lo complicada que puede llegar a ser la comunicación hacia otros y hacia ellos mismos, disminuyendo su calidad de vida, siendo la educación una de las principales y primeras áreas afectadas en la vida de quien padece esta discapacidad [3], teniendo en ocasiones que solicitar tutores sombra, los cuales se encargan de traducir o comunicar lo visto en clase, pero dado que esta ayuda aún no es obligatoria en las escuelas (educación preescolar y básica), es muy difícil y costosa conseguirla, y a lo mucho se obtiene este tipo de ayuda hasta nivel medio superior, quedando prácticamente a la deriva, y si alguna persona Sorda decidiera iniciar una carrera profesional en alguna Universidad, tendría dificultad para poder ingresar, mantenerse en ella y conseguir el grado al final de la misma [4].

## **1.3. Preguntas de Investigación**

- ¿Cuál es la estructura de una frase correcta en LSM?
- ¿Cuál es el diseño de un módulo de Razonamiento Basado en Casos mediante un SMA?
- ¿Cuál es el diseño de un modelo basado en un SMA que realice un proceso de razonamiento para buscar frases en LSM utilizando CBR?

## **1.4. Objetivos**

Con la finalidad de desarrollar un sistema que ayude a tener una mejor comunicación a personas Sordas, principalmente en el ámbito de la educación y

con las personas que las rodean, que suelen ser familiares y amigos, se establecieron los siguientes objetivos.

#### **1.4.1. Objetivo General**

Diseñar un modelo basado en un SMA que realice un proceso de razonamiento, para buscar frases en LSM utilizando CBR.

#### **1.4.2. Objetivos Específicos:**

- Estudiar las diferentes temáticas que inciden en el estado del arte.
- Investigar sobre gramática que determina la LSM.
- Definir la arquitectura propuesta del SMA.
- Modelar un SMA que permita reconocer frases en LSM mediante un módulo de razonamiento utilizando CBR.

### **1.5. Justificación**

Dado que en México hay pocos traductores certificados, se propone diseñar un modelo Multi-Agente que permita reconocer frases LSM utilizando CBR para el apoyo de la comunidad Sorda, principalmente en la vida cotidiana y el ámbito escolar, donde se presentan grandes dificultades, como en la familia, en la sociedad, en el trabajo y en general cualquier ambiente en donde se necesite constante comunicación entre personas.

Lograr terminar este proyecto, contribuirá con una propuesta que permita implementar el modelo diseñado en un sistema de reconocimiento de voz y traducción a LSM y facilitar la convivencia en sociedad con las personas Sordas. Esto último, plantea el reto de apoyar las personas sordas una mejor convivencia con quienes les rodean.

## **1.6. Alcances y Delimitaciones**

Los resultados que se pretenden alcanzar al tener este proyecto, así como sus limitaciones son las siguientes:

- Los estudios a realizar serán apoyados por la comunidad de Sordos de la ciudad de Hermosillo Sonora.
- Sólo se trabajará con la gramática en LSM.
- El resultado final es un modelo Multi-Agente que se presenta como propuesta y que ayudará a las personas Sordas a tener una mejor convivencia con quienes les rodean.

## **Capítulo 2: Estado del Arte**

En este capítulo se presenta la información donde se describen las herramientas y conceptos utilizados dentro de este proyecto, así como temas derivados o relacionados de los mismos, necesarios para comprender mejor la función de lo que aquí se presenta, obtenidas de diversas fuentes de las bases de datos proporcionadas por las entidades escolares.

Los principales temas a tratar son la Lengua de Señas Mexicana (LSM) y las personas Sordas, temas a los que va dirigida principalmente esta investigación. En esta sección se verá de manera general en qué consisten estos términos y cómo surgieron.

Posteriormente se inicia con la descripción de las herramientas a utilizar, estas herramientas son los softwares o plataformas, sugeridas durante el desarrollo del proyecto, donde se enumeran y señalan las características principales de los mismos y a su vez se enfatiza la descripción de la o las herramientas adecuadas en este caso.

Seguidamente se muestran las técnicas a utilizar en conjunto con nuevas tecnologías, como lo son el uso de Agentes Inteligentes en los Sistemas Multi-Agentes (SMA), Razonamiento Basado en Casos (CBR, por sus siglas en inglés), donde nuevamente se da una descripción explicando el uso de las mismas, detallando algunos de sus principales aspectos y características que éstas presentan.

Por último, se mencionan algunos de los trabajos relacionados a esta tesis.

## **2.1. Lengua de Señas Mexicana**

La comunicación es fundamental para el desarrollo social del ser humano. Existen diversas formas de comunicación, entre ellas la manera de expresión más común es la oral y acompaña a las personas como herramienta de participación durante toda su vida. En el momento que esta forma de comunicación se ve limitada o impedida, la realización social se reduce de manera significativa. Las personas Sordas tienen dificultad para comunicarse lo que disminuye su capacidad de interacción social; en consecuencia, su desarrollo educativo, profesional y humano queda restringido seriamente.

Como medio de socialización de personas sordas, se ha ido desarrollando su propia lengua, la lengua de señas, pero aun cuando esta medida es efectiva entre personas que la utilizan, para el resto sigue siendo un impedimento para la comunicación [10].

La lengua de señas es un lenguaje visual utilizado principalmente por personas Sordas. Este tipo de lenguas, a diferencia del lenguaje natural, se basa en símbolos o gestos realizados mediante señas, los cuales pueden representar desde letras individuales hasta frases, es una manera de comunicación, que varía dependiendo del contexto en el que se esté hablando. Esta lengua no es universal, cada país o región tiene su propia lengua de señas, en este caso la lengua de señas de México, es la LSM, la cual se basa en la lengua de señas francesa.

Según varios estudios realizados, las personas con alguna discapacidad auditiva, mostraron deficiencia al leer, lo cual demuestra la posible relación entre la lectura y la deficiencia auditiva, gracias a esto la comunicación hacia las personas Sordas se puede volver más complicada [11].

El Sordo no es “mudo”, pero no es capaz de utilizar medios orales de comunicación, así, este impedimento estigmatizó al Sordo. Según Oliver Sacks, la sordera congénita se da en todas las razas y países, y así ha sido desde el principio de la historia. Afecta a una milésima parte de la población. Lo crucial es

nuestro conocimiento de los Sordos y nuestra actitud hacia ellos, la comprensión de sus necesidades específicas, el reconocimiento de sus derechos humanos fundamentales, como el acceso sin restricciones a un idioma natural y propio, a la enseñanza, el trabajo, la comunidad, cultura, educación, a una existencia plena e integrada.

Unos de los antecedentes que se pueden mencionar sobre la educación del sordo en México, es la creación de la Escuela de Sordomudos y la fundación de la Escuela Nacional de Sordomudos como escuela normal, ambas creadas por decreto presidencial de Benito Juárez [12].

## **2.2. Razonamiento Basado en Casos**

El CBR (Por sus siglas en inglés, Case Based Reasoning), es una técnica de inteligencia artificial que se basa en almacenar ejemplos a partir de los cuales se pueden proponer soluciones a nuevos problemas, reutilizando la experiencia acumulada hasta el momento, para así obtener una nueva solución a un problema, la cual será almacenada para usos futuros. El razonamiento basado en casos, trata de imitar la manera de llegar a soluciones de los seres humanos.

En la actualidad el CBR ha empezado a desempeñar un papel importante en la ingeniería de software, se ha ido incrementando el uso de esta técnica en numerosas aplicaciones, por lo que se eleva notablemente la calidad del proceso de desarrollo de software.

En la actualidad existen muchas maneras de realizar aprendizaje automático, área que pertenece a la inteligencia artificial.

Dentro del aprendizaje automático se pueden mencionar diversas técnicas, tales como redes neuronales, aprendizaje inductivo, aprendizaje adaptativo, lógica difusa y CBR, diferenciándose entre ellas por la forma en que el conocimiento se almacena, procesa, se toman decisiones y se realizan acciones.

En el caso del CBR el conocimiento está formado por una base de creencias, donde se encuentran problemas resueltos en el pasado [13], de alguna manera tratando de imitar la manera en que las personas cuentan con experiencias pasadas, con las cuales pueden resolver problemas futuros.

A continuación, se muestra un diagrama general del funcionamiento de un Sistema de Razonamiento Basado en Casos (SRBC) en la Fig. 2.1.

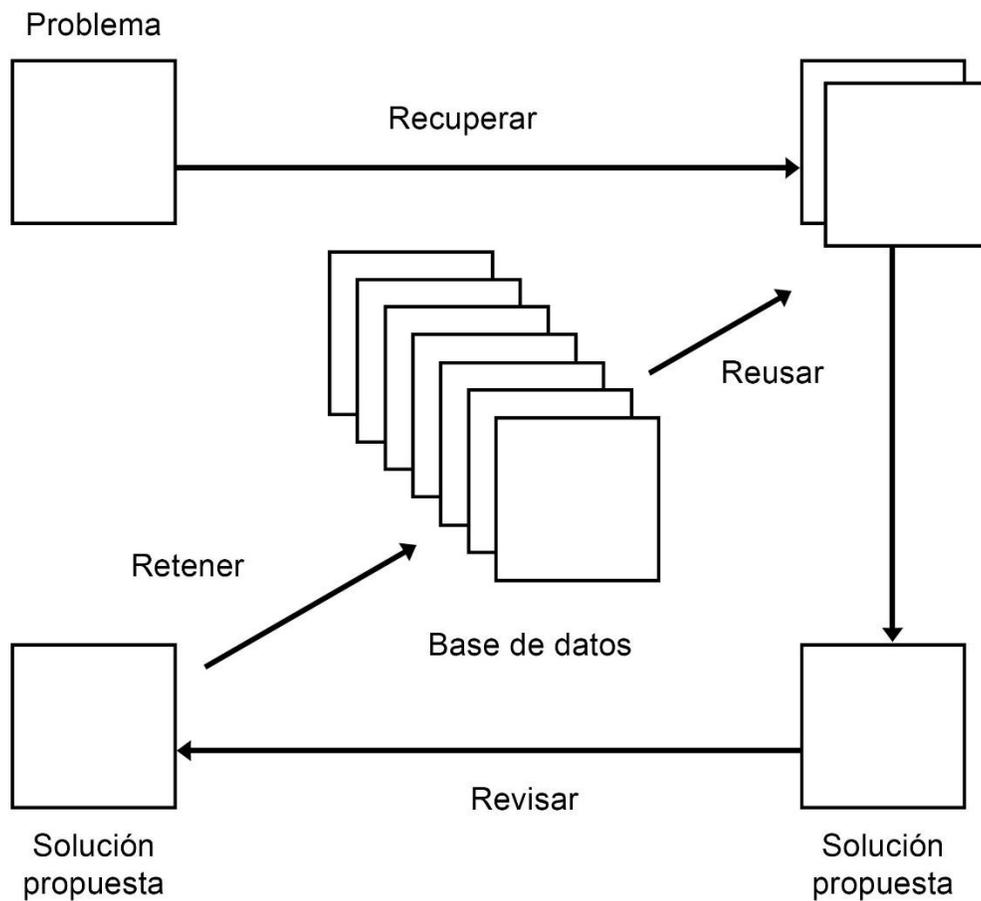


Figura 2.1 Diagrama CBR

Como se muestra en la figura 2.1, el funcionamiento del CBR inicia en el problema, recuperando los casos más parecidos, donde reutiliza los casos para

resolver el nuevo problema, los adapta para el nuevo problema y almacena la solución como parte de un nuevo caso. Un caso mantiene todos los atributos y características relevantes de un evento pasado, mismas que servirán como índices para la recuperación de caso futuro [14].

Básicamente, el CBR es la resolución a problemas utilizando soluciones a situaciones similares a problemas anteriores, en pocas palabras reutiliza el conocimiento y la información según sea conveniente.

Existen varios tipos de métodos CBR, pero los principales son:

- Razonamiento Basado en Ejemplares:
  - Son sistemas CBR que se centran en aprendizaje de definiciones de conceptos.
  - Un concepto se define por su extensión: el conjunto de ejemplares.
  - El sistema se limita a realizar clasificación, asignándole una categoría al caso nuevo. Sin adaptación.
- Razonamiento Basado en la Memoria:
  - Se hace énfasis en la organización y en el acceso a grandes memorias de casos.
  - Uso de técnicas de procesamiento paralelo.
- Razonamiento Basado en Casos:
  - Cierta complejidad en la presentación de los casos.
  - Incluyen mecanismos de adaptación.
  - Incluyen algún tipo de conocimiento de propósito general.
- Razonamiento Basado en Analogías:
  - Intentan resolver un nuevo problema utilizando casos antiguos provenientes de un dominio de conocimiento diferente [15].

Los CBR presentan algunas ventajas frente a otros sistemas tradicionales:

- **Adquisición de conocimiento:** La adquisición de conocimiento se realiza a partir de las experiencias previas almacenadas en la base de creencias o librería de casos.
- **Mantenimiento del conocimiento:** Los CBR permiten el incremento de nuevos casos a la base de creencias, sin la intervención del experto, haciendo innecesario el proceso de mantenimiento de la misma.
- **Eficiencia en la resolución de problemas:** La reutilización es un principio básico en informática. Los CBR permiten que se puedan resolver casos similares sin tener que rehacer la base de conocimientos.
- **Calidad de la solución:** Al aplicar el principio de optimalidad, se garantiza memorizar la mejor solución o lo que ha sucedido en un contexto determinado.
- **Aceptación del usuario:** Utilizar soluciones basados en casos que ya han sido utilizados y probados da confianza al usuario, lo que no ocurre en sistemas con redes neuronales, ya que los resultados pueden llegar a ser incomprensibles.

Los Sistemas de Razonamiento Basado en Casos (SRBC) se pueden utilizar en situaciones donde se cumple:

1. Los dominios son difíciles de estructurar, por tanto, es difícil establecer un modelo del dominio.
2. Situaciones en donde resulta difícil explicar la solución de un problema.

Los SRBC también tienen restricciones como lo son:

1. El dominio de la aplicación no debe ser cambiante. Lo que es cierto hoy, deberá serlo mañana.
2. Los problemas deben repetirse en algunas ocasiones, de no ser así y solo tratarse de problemas de una sola o pocas ocasiones, no será necesario memorizar un caso nuevo.

Los métodos de SBRC deben procesar ejemplos existentes para hacer comparaciones entre casos nuevos y viejos. Para realizar estos procesos existen tres maneras de hacerlo.

- Memorizar todos los ejemplos, pero causa retardos.
- Memorizar una parte, eligiendo solo los ejemplos más importantes.
- Creación de prototipos, lo que representa conjunto de datos ficticios.

Para la solución de un SRBC se requieren los siguientes pasos:

- Crear una estructura para almacenar los problemas a resolver.
- Crear una estructura con las soluciones.
- Se diseña una solución para resolver el caso.

La solución consta de dos fases:

- Recuperación de casos.
- Selección del caso más similar al problema.

Dentro de un SRBC, el “Caso” es el elemento principal, en el cual se encuentra la descripción del problema y la solución del mismo, mismo que pudo existir desde el inicio del sistema o haber sido añadido o creado por el mismo sistema posteriormente.

Una librería de casos, es similar a una base de datos que organiza los casos de manera estructurada. La realización más sencilla de una Librería de Casos es mediante una memoria plana, pero puede utilizarse también mediante memoria jerárquica.

Para la realización de un SRBC se puede proponer un algoritmo como el que se muestra a continuación:

## **Acción SRBC()**

### **Inicio**

**RECORDAR** los casos similares al analizado.

**REUTILIZAR** la información y el conocimiento que se tiene para resolver el problema.

**REVISAR** la solución sugerida.

**RETENER** las experiencias que puedan ser útiles para la resolución de futuros problemas.

### **Fin.**

Dentro de un SRBC el primer paso a realizar al tener un problema, es recordar todos los casos relevantes que pudieran ser de ayuda para resolver ese problema. Este es uno de los principales problemas, pues se tendrán que recuperar los casos seleccionados de la base de conocimientos. Después de la recuperación de los casos, se tendrán que adaptar para la solución que se busca ya sea en su totalidad o parcialmente dependiendo del presente problema.

Con esto resuelto, se tendría una primera versión de la posible solución al problema, pero aún queda probar esta solución encontrada, ya sea en el mundo real o en una simulación y de ser necesario revisarla. Quedando todo este proceso en un ciclo de selección, adaptación y revisión de los casos, y, por último, una vez teniendo la solución al nuevo problema, el último paso es la retención del caso, almacenándolo en la Librería de casos [16].

## **2.3. Agentes Inteligentes**

Los agentes inteligentes constituyen un paradigma de programación para el desarrollo de aplicaciones software. La creación de nuevos paradigmas de programación, ha sido uno de los mayores retos para los ingenieros en el siglo XXI. Los agentes son un fuerte candidato dentro de estos paradigmas.

Los agentes inteligentes cuentan con diversas funciones y características, de las cuales se pueden destacar:

- **Autonomía:** Se refiere a la capacidad del agente, el cual es capaz de realizar acciones autónomas en un determinado ambiente para comprender sus objetivos.
- **Sociabilidad:** Los agentes cuentan con una habilidad social, lo que les permite interactuar con otros agentes y posiblemente humanos u otra entidad.
- **Reactividad:** Los agentes inteligentes pueden percibir su ambiente alrededor y responder en tiempo a los cambios ocurridos en el mismo.
- **Proactividad:** Es la capacidad de ir directamente por su objetivo tomando la iniciativa.
- **Benevolencia:** Es la comunicación entre agentes para cooperar entre ellos, siempre y cuando no intervenga con sus objetivos.
- **Racionalidad:** El agente siempre realiza la acción correcta de acuerdo a su entorno y experiencia.

Aparte de estas características de los agentes, estos también pueden clasificarse según aspectos que los diferencian unos de otros. Considerando estos aspectos, los agentes se pueden clasificar en:

- **Agentes reactivos:** Se caracterizan por realizar tareas sencillas fundamentadas en la recepción de eventos externos. El comportamiento de estos, puede ser un cambio de estado interno o la ejecución de funciones sobre el entorno. Estos agentes no realizan procesos de razonamiento ni representan conocimiento.
- **Agentes cognitivos:** Realizan tareas complejas realizando procesos de razonamiento, planificación y aprendizaje. Se basan en el modelo computacional Percibir →Asimilar → Razonar →Actuar.

Existe una clasificación de agentes también por su organización que pueden ser:

- **Agentes individualistas:** No necesitan de otros agentes para lograr sus objetivos. Trabajan solos y no pueden cooperar.
- **Agentes cooperantes:** Son capaces de realizar tareas tanto solos como con otros agentes.

De acuerdo a su aplicación, también se pueden clasificar en:

- **Agente de interfaz o usuario:** El agente puede funcionar como asistente personal y tener características de aprendizaje y autonomía. Se utilizan para enseñarle o servirle de guía o apoyo al usuario en el manejo de alguna aplicación.
- **Agente de búsqueda:** Construye sus acciones a partir de trozos de información. No se trata de solo un algoritmo de búsqueda y debe dar buenos resultados al usuario.
- **Agente de monitoreo:** Son agentes que se ejecutan constantemente, vigilando las entradas, salidas y modificaciones dirigidas hacia algún sistema, registrando y dando aviso de los eventos del mismo al usuario u otro agente del sistema.
- **Agente de filtrado:** Trabaja en conjunto con el agente de monitoreo, este agente mantiene la información actualizada para el usuario.

Unas de las características de los agentes que los hacen diferenciarse de los objetos, es que estos, cuentan con una noción más fuerte de autonomía, ellos pueden decidir si realizar o no una acción. Los agentes también cuentan con un comportamiento flexible basado en su reactividad, proactividad y sociabilidad. Por último, en un SMA, en esa situación, cada agente cuenta con al menos una tarea asignada que realizar [17].

Un agente puede ser visto como un robot que percibe el ambiente mediante sensores y actuadores como se ve en la figura 2.2, igual como lo haría un humano a través de sus sentidos (Vista, oído, olfato, gusto, tacto). Un agente robótico tendría cámaras, sensores de temperatura, humedad, infrarrojos y varios motores como actuadores.

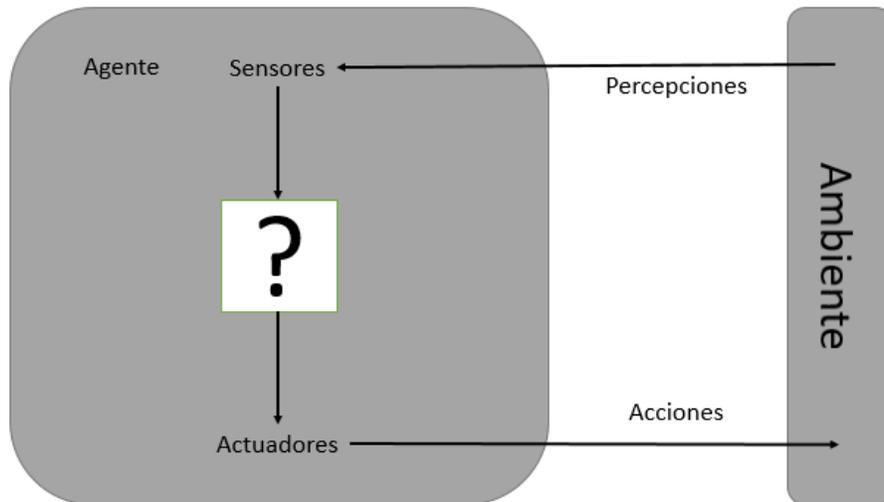


Figura 2.2 Diagrama Funcionamiento Agente

Para poder diseñar un agente se tienen que especificar diferentes aspectos de las tareas que se necesita que el agente realice, las cuales consisten en el desenvolvimiento que el agente debe tener, el ambiente en el que se va a situar, los actuadores con los que el agente va a contar y los sensores necesarios [18]. A continuación, se muestra un ejemplo de estos aspectos en la siguiente Tabla 2.1.

Tabla 2.1 Ejemplo de aspectos de un agente

Tipo de Agente	Desenvolvimiento	Ambiente	Actuadores	Sensores
Conductor de taxi	Seguro, rápido, legal, cómodo, eficiente	Calles, tráfico, transeúntes, señalamientos	Volante, acelerador, freno, luces	Cámara, velocímetro, odómetro, sensores de motor, GPS

-Desenvolvimiento del agente: Se refiere al comportamiento que el mismo tendrá durante la tarea, sus metas, cualidades, decisiones, etc., en el caso del ejemplo de la tabla, una conducción segura, legal, cómoda y eficiente.

-Ambiente: Se trata del ambiente en el que se desenvolverá el agente, en el caso del ejemplo de la tabla, las calles de una ciudad, el tráfico, los transeúntes, los señalamientos, etc.

-Actuadores: Son las herramientas que el agente utilizará dependiendo de la decisión tomada, las mostradas en el ejemplo son el volante, el freno, el acelerador, etc.

-Sensores: Son las herramientas que estarán en constante monitoreo del ambiente, como pueden ser cámaras, velocímetro, odómetro, GPS, sensores del motor, etc.

Para saber qué tipo de tareas necesitará llevar a cabo un agente o agentes, es necesario identificar el tipo de entorno donde se encontrará, teniendo en cuenta estas características [18]:

- **Completamente observable o parcialmente observable**: Si los sensores de un agente son capaces de medir el estado del ambiente en cualquier punto en cualquier momento, entonces decimos que el entorno es completamente observable. En cambio, si las mediciones de los sensores son incompletas o presentan interferencias, se puede decir que el entorno es parcialmente observable.
- **Agente único o multi-agente**: En este caso la diferencia entre ambos puede ser simple. Por ejemplo, en un juego individual de cartas como sería el solitario, el jugador solo es uno, en este caso el agente, el cual estaría tomando las decisiones óptimas para el juego, pero no estaría interactuando con algún otro actor, en el caso de un sistema multi-agente, se puede poner como ejemplo un juego de ajedrez, donde los actores son dos, que interactúan con el entorno, siendo cada actor un agente diferente en un ambiente competitivo. Otro ejemplo sería el caso de un conductor de taxi, donde el entorno sería cooperativo debido al ambiente donde se desenvolverían los agentes.
- **Determinístico o estocástico**: Si el siguiente estado del entorno está completamente determinado por el estado actual y la acción realizada por el

agente, entonces decimos que se trata de un ambiente determinístico, en caso contrario, es estocástico.

- **Episódico o secuencial:** En un ambiente episódico el agente no se tiene que preocupar por cambios en el ambiente por las decisiones tomadas, en cambio un ambiente secuencial el ambiente puede cambiar según las decisiones tomadas por los agentes a largo plazo.
- **Estático o dinámico:** Si el ambiente puede cambiar mientras el agente se encuentra deliberando, entonces el ambiente es dinámico para el agente, en caso contrario, es estático. En el caso que el ambiente no cambie con el pasar del tiempo, pero que una acción del agente influya y pueda cambiar el mismo, se le determina como semidinámico.
- **Discreto o continuo:** Esta característica aplica para el estado del entorno, el cual se ve afectado con el tiempo, con las percepciones y acciones de los agentes. Mientras los estados del entorno sean finitos el entorno se considera discreto, pero si la cantidad de estados no puede contabilizarse se considera continuo.
- **Conocido o desconocido:** En este caso se trata de entornos en los cuales el agente o el diseñador está consciente o no del mismo. En los casos donde el ambiente sea conocido, el agente contará con las acciones necesarias para interactuar con este, en caso contrario el agente deberá a aprender a como desenvolverse en el entorno.

A continuación, se muestra una tabla 2.2 con ejemplos con las características antes mencionadas:

*Tabla 2.2 Ejemplo de entornos y agentes*

Entorno	Observable	Agentes	Determinista	Episódico	Estático	Discreto
Ajedrez	Completo	Multi	Determinista	Secuencial	Semi	Discreto
Poker	Parcial	Multi	Estocástico	Secuencial	Estático	Discreto

Conductor de taxi	Parcial	Multi	Estocástico	Secuencial	Dinámico	Continuo
Análisis de imagen	Completo	Único	Determinista	Episódico	Estático	Continuo
Tutor de inglés	Parcial	Multi	Estocástico	Secuencial	Dinámico	Discreto

[19]

### 2.3.1. Sistemas Multi-Agentes

Un Sistema Multi-Agente (SMA), es un sistema compuesto por múltiples elementos computacionales interactuando, mejor conocidos como agentes [19]. Añadiendo a lo anteriormente mencionado, los agentes son sistemas computacionales con dos importantes funciones. Primero, ellos son capaces de realizar acciones de manera autónoma y en segundo, ellos son capaces de interactuar con otros agentes, no solamente intercambiando información, sino interactuando entre ellos como lo haría una sociedad en el día a día.

En un SMA es necesaria la comunicación, para así conocer de mejor manera su entorno, para que, al interactuar los agentes entre ellos, se puedan resolver los objetivos generales dentro de un sistema. Los puntos de vista de cada agente son limitados y los datos están descentralizados, por ello, la comunicación es esencial. La implementación de un SMA es asincrónica, lo que quiere decir que los agentes no actúan todos a la vez.

Los SMA cada vez son más utilizados gracias a su similitud de su comportamiento con el de una sociedad, lo que vuelve al sistema, más entendible en su funcionamiento, pero que a la vez son de gran ayuda para integrarlos en sistemas complejos dependiendo de la aplicación donde se vayan a utilizar [19].

### 2.3.2. Arquitecturas Multi-Agentes

Cada agente cumple una tarea o función diferente a otro, dependiendo de sus características, habiendo así agentes más complejos o inteligentes que otros. Por ello, la arquitectura seleccionada dependerá de las características de los agentes, las tareas y del entorno donde se desenvolverán.

Una arquitectura determina los mecanismos que utiliza o utilizan los agentes para realizar sus funciones y/o comunicarse entre ellos, con el fin de que el agente logre determinada tarea. La diferencia entre arquitecturas se da dependiendo de la forma en que se interconecten los módulos que componen los agentes, así como también el método de descomposición de las tareas.

En el ámbito de los agentes, se pueden mencionar gran variedad de arquitecturas, las cuales se pueden clasificar como [20]:

- **Reactivas:** Las caracteriza el no tener un modelo simbólico como elemento central de razonamiento y no utilizan un razonamiento simbólico complejo.
- **Deliberativas:** Utilizan modelos de representación simbólica del conocimiento, es decir, contiene un modelo simbólico del mundo, donde se decide a través de mecanismos de razonamiento lógico basados en correspondencia de patrones y manipulación simbólica.
- **Híbridas:** Combina las arquitecturas reactivas y deliberativas, reduciendo así, las limitaciones que estas podrían tener.

#### 2.3.2.1. Arquitecturas Reactivas

Arquitectura de subsunción: Es una de las mejores arquitecturas reactivas, puesto que está basada en tres propuestas [21]:

1. El comportamiento inteligente puede ser generado sin la representación explícita del tipo de la inteligencia artificial simbólica.
2. El comportamiento inteligente puede ser generado sin el razonamiento abstracto de la inteligencia artificial simbólica.

3. La inteligencia es una propiedad emergente de algunos sistemas complejos.

PENGI [21]: El autor de esta arquitectura, observó que cualquier actividad diaria está compuesta por rutinas, que solo cambiaban un poco entre ellas dependiendo de la tarea que se requiera realizar. Al aprender todas las tareas, estas pueden realizarse mediante rutinas, solo con pequeñas variaciones. Pero esta idea, al solo contener rutinas y no ser aptas para nuevos comportamientos, pueden ser programadas con lenguaje de bajo nivel o incluso dentro del circuito de algún sistema, donde solo se actualizarán periódicamente.

Autómata situacional [21]: Trata de un paradigma de agentes inteligentes, que reacciona en base a sus conocimientos y metas. Son agentes especificados en términos de dos componentes: percepción y acción. Este tipo de paradigma ha atraído mucho la atención, porque combina lo mejor de los sistemas reactivos y simbólicos.

La arquitectura de agentes en red: Pattie Maes desarrolló una arquitectura de agentes en donde los agentes son definidos como módulos competentes [21]. Cada módulo se diseña en base a pre y pos condiciones, y un nivel de activación, el cual da una indicación de valor real de la relevancia del módulo en una situación particular. Entre mayor sea el valor de activación del módulo, mayor será el impacto del mismo en el agente. Este tipo de arquitectura, es similar a las redes neuronales.

Dentro de las arquitecturas propuestas de agentes reactivos, existen obvias ventajas, pero también existen varios problemas fundamentales sin resolver, como pueden ser:

- Si los agentes no implementan modelos en base a su ambiente, entonces ellos deberán tener la suficiente información sobre este para poder determinar una acción aceptable.

- Puesto que los agentes reactivos reaccionan solo en base a información local, difícilmente podrán tomar alguna decisión cuando la situación presente no sea local.
- Mientras que los agentes efectivos son desarrollados con una pequeña cantidad de comportamientos, se vuelve mucho más complicado manejarlos cuando estos cuentan con varios de estos, volviendo la interacción de los diferentes comportamientos muy complicados de comprender [18].

### **2.3.2.2. Arquitecturas Híbridas**

PRS (Procedural Reasoning System) [21]: Uno de las mejores arquitecturas de agentes es la del Sistema de Procesado de Razonamiento, desarrollado por Georgeff y Lansky. Esta arquitectura está basada en las arquitecturas BDI, pero incluyendo librerías y símbolos explícitos, que representan las partes de una arquitectura BDI.

Máquina de Turing: Una arquitectura que consiste en la percepción y acción de subsistemas, que interfieren directamente con el agente de entorno, y tres capas de control embebidas en un framework de control, que se maneja entre las capas. Cada capa es independiente de otra [21].

### **2.4. Animaciones Mediante Avatar**

Cada vez los personajes virtuales se vuelven más comunes al usar dispositivos como computadoras, tabletas o celulares, para la comunicación usuario máquina, y para este caso, el uso de un avatar se vuelve esencial para poder traducir las frases en señas [22].

En Google se puede encontrar la definición de avatar como, un icono o figura representativa de una persona en videojuegos, foros de internet, blogs, etc. El uso

de los avatares se puede presentar principalmente en videojuegos, pero también en ambientes educativos o comerciales.

#### **2.4.1. Opciones de Diseño para un Avatar**

Existen numerosas opciones para la creación de avatares, incluso se ha desarrollado software específico para el diseño de los mismos, pero los mejores resultados se obtienen con software especializado en diseño 3D.

*Blender*, es un software profesional para la creación y edición de objetos en 3D, el cual también permite realizar animaciones, videojuegos, simulaciones físicas, entre otras características [9]. En este caso, para el diseño del avatar, constará de 3 procesos, modelado del avatar, proporcionarle articulaciones y animarlo, estos tres procesos pueden ser realizados por la herramienta antes descrita.

Para el primer proceso es necesario contar con modelos 3D, los materiales y utilizar bien las luces, y utilizar estos componentes para tratar de imitar el objeto o persona del mundo real.

Con el proceso de modelo concluido, se tendrá un avatar inmóvil, por lo tanto, lo siguiente es articular al modelo, una de las herramientas sugeridas es la de *Armature*, la cual permite articular el modelo creado como si de huesos se tratasen, para poder darle movimiento, limitando los mismos por el posicionamiento de las articulaciones creadas.

Por último, para el proceso de animación, existen distintas técnicas para lograrlo, una de ellas es el *Motion Track* o rastreo de movimiento [9], un dispositivo que utiliza esta técnica, es la herramienta de Windows “*Kinect*” utilizada originalmente en la consola de videojuegos Xbox. Esta técnica puede ser utilizada con cualquier tipo de cámara digital, aunque dependerá mucho de la misma, siendo que, en este caso, será de mayor funcionalidad una que pudiera proporcionar profundidad, característica con la que cuenta el dispositivo “*Kinect*” antes mencionado, aunque existen muchas otras alternativas con diferentes sensores que proporcionan esa misma característica tan importante [9].

## **2.5. Trabajos Relacionados**

Existe poco material relacionado a trabajos que involucren deficiencia auditiva, agentes inteligentes, razonamiento basado en casos y realidad aumentada. A continuación, se enlistarán los trabajos más relevantes al tema de investigación de los que se tiene conocimiento.

### **2.5.1. Chinese Sign Language Animation System On Mobile Devices**

Es un trabajo realizado en china, donde se explica la problemática que tienen en su país las personas con dificultades auditivas y del habla, por lo que proponen el desarrollo de aplicaciones móviles donde se muestra un avatar en 3D que traduce del chino al lenguaje de señas chino. En el trabajo se menciona que existen numerosos proyectos en diferentes países donde se han estado haciendo avances en estas áreas para personas discapacitadas, pero realizadas con tecnologías diferentes o enfocadas a otros aspectos.

En este proyecto el grupo de personas a las que está dirigido principalmente, es a jóvenes de entre 1 y 14 años, que es cuando las dificultades de comunicación se hacen más evidentes, recalcando que entre dos personas sordas no existe mayor problema de comunicación al poderse los dos comunicar con el lenguaje de señas, pero al tratarse de personas que no presentan esa condición es cuando el problema se vuelve evidente.

En el documento se puede apreciar cómo se va describiendo a detalle el funcionamiento de la aplicación sugerida y de las herramientas utilizadas para su creación, así como su arquitectura y posible funcionamiento en algunas plataformas [23].

### **2.5.2. IBM System Is a Virtual Sign-Language Interpreter**

Al igual que en el trabajo anteriormente mencionado, este proyecto trata sobre el uso de un avatar para traducir de voz a texto a lenguaje de señas, pero en este

caso de Inglaterra. En este trabajo se utilizan tecnologías de reconocimiento de voz la cual filtra los diálogos y crea oraciones con la gramática adecuada y de ahí se basa para la creación de las señas, las cuales son representadas por un avatar animado. Este proyecto dice ser de apoyo para los estudiantes con problemas de audición principalmente, pero que no pretende reemplazar a los intérpretes humanos, si no para servir de apoyo en casos en los que no se tenga la ayuda de un intérprete.

A diferencia de estos trabajos mencionados, en el trabajo propuesto en esta tesis, se pretende realizar un proyecto que decida mediante inteligencia artificial la selección de las señas a utilizar al momento de realizar la traducción y así reducir tiempo de procesamiento, al almacenar los casos reutilizables dentro de una base de creencias [24].

### Capítulo 3: Análisis y Diseño

En el presente capítulo se ve el diseño del SMA para el reconocimiento de frases LSM utilizando CBR.

Así mismo, se presentan los diferentes modelos para el desarrollo de un SMA en conjunto con un SRBC, con la finalidad de obtener una frase formada en LSM. Para ello se utilizarán diferentes herramientas, tales como JADE, para la programación del sistema dentro de la plataforma NetBeans y la metodología INGENIAS, utilizando un IDK desarrollado para la creación de este tipo de modelos.

En este capítulo se definen los diagramas de contexto de nivel 0 y los diferentes modelos que representan un SMA utilizando CBR, para la creación de representaciones para Lenguaje de Señas Mexicano.

#### 3.1 Diagrama de Contexto

En los siguientes puntos, se ven los diagramas que definen el sistema, desde una perspectiva muy general hasta una más específica, en los cuales se dará una breve explicación resumiendo y explicando lo que se muestra en cada uno de estos diagramas.

##### 3.1.1 Diagrama de Contexto de Nivel 0

En la siguiente Figura 3.1 se muestra el diagrama de contexto nivel 0 del SMA para el reconocimiento de frases LSM utilizando CBR. En el diagrama se presenta el proceso del sistema en cuanto a la frase ingresada, procesada y proyectada como resultado final, de una manera simple y general.



Figura 3.1 Diagrama de Contexto Nivel 0

### 3.1.2 Diagrama de Contexto de Nivel 1

En la siguiente Figura 3.2 se muestra un diagrama de contexto nivel 1, mostrando con mayor detalle, al diagrama anterior, el funcionamiento del sistema. En este diagrama se muestra principalmente el proceso del CBR para la selección y creación de nuevos casos, siendo esta la sección más importante del sistema puesto que es donde se recibe la información y donde se obtiene un resultado, para ser enviado. También se muestran a los encargados de enviar y recibir la frase y el resultado de la misma, los cuales serían el Agente Coordinador y Agente Interfaz, último que también se encargaría de la proyección del resultado en la interfaz utilizada.

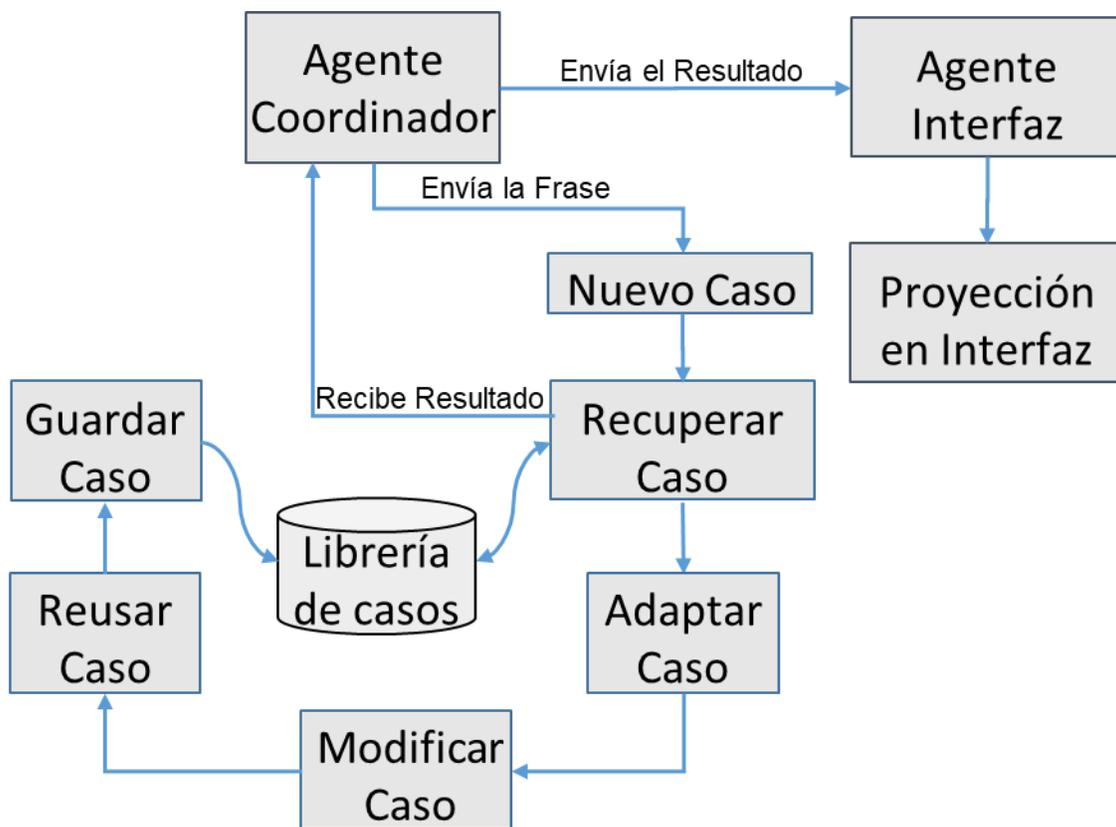


Figura 3.2 Diagrama de Contexto Nivel 1

### 3.2. Arquitectura del Sistema Propuesto

El esquema mostrado en la Fig. 3.3. representa la arquitectura propuesta del modelo del sistema Multi-Agente para reconocer frases en LSM utilizando CBR.

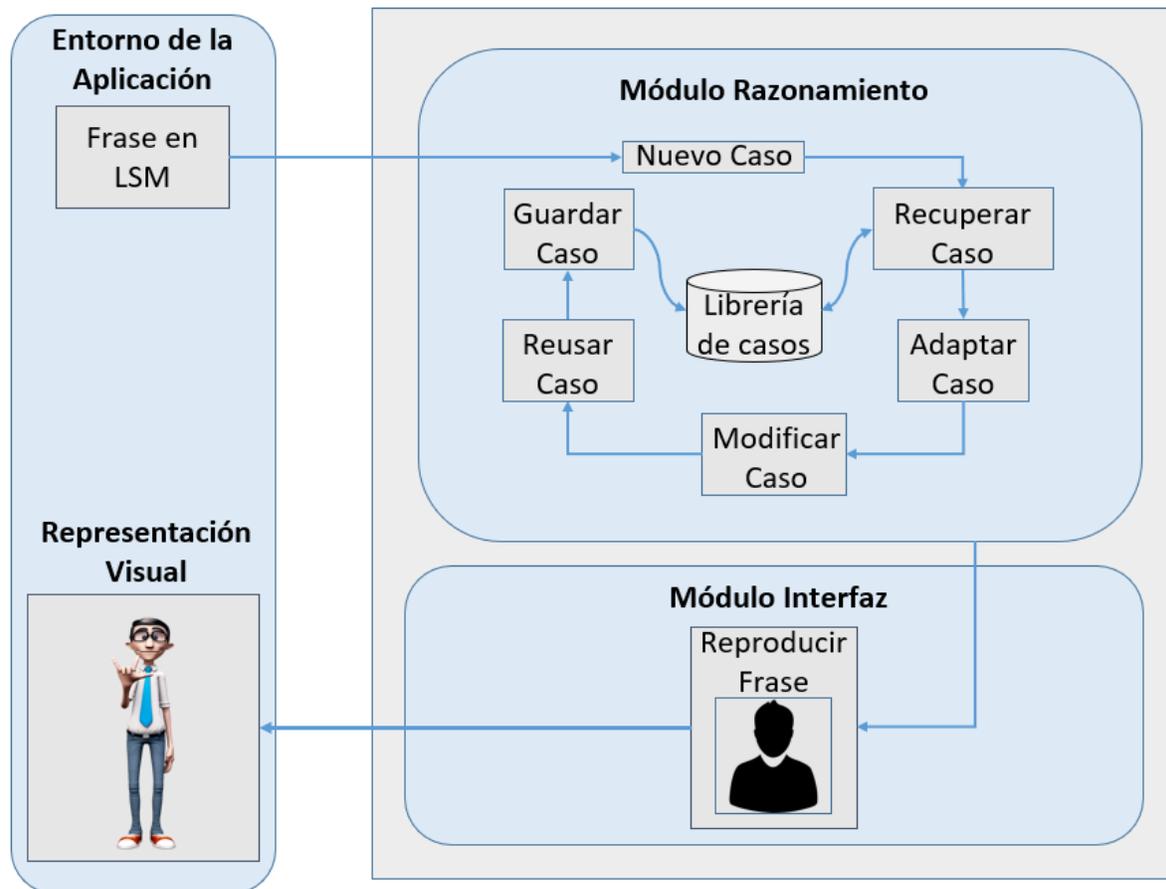


Figura 3.3 Arquitectura Propuesta

A continuación, se describen cada uno de los módulos que se indican en la arquitectura propuesta.

- Entorno de la Aplicación: es donde el usuario interactuará con el SMA propuesto, el cual propone las frases en LSM y en donde se puede observar el resultado obtenido al procesar la frase y representarla de manera visual.
- Módulo de Razonamiento: es donde se realizará el proceso de recibir la frase en LSM y revisar si dicha frase está en la base de creencias

predefinida con frases en LSM comunes, tomando la decisión de crear el caso en la base de creencias si ésta no existe o presentar una solución como sugerencia para reproducirla más adelante. Resumiendo, este módulo se encarga de analizar si la frase existe en la base de creencias, modificar o crear el nuevo caso.

- Módulo Interfaz: Finalmente, una vez obtenido el caso, este será enviada al último módulo, el cual se encargará de reproducir la animación para ser proyectada en el entorno de la aplicación.

### **3.3. Diagramas**

En esta sección se muestran los diferentes diagramas que describen las partes del sistema y el funcionamiento del mismo.

#### **3.3.1. Casos de Uso**

Para modelar las capacidades del SMA propuesto, se han identificado tres casos de uso principales: Obtener Frase, Reconocer Frase y Proyectar Frase. Así como el actor encargado de tomar la decisión, denominado Agente Coordinador y el agente encargado de recibir la frase, denominado Agente Interfaz como se muestra en la Fig. 3.4.

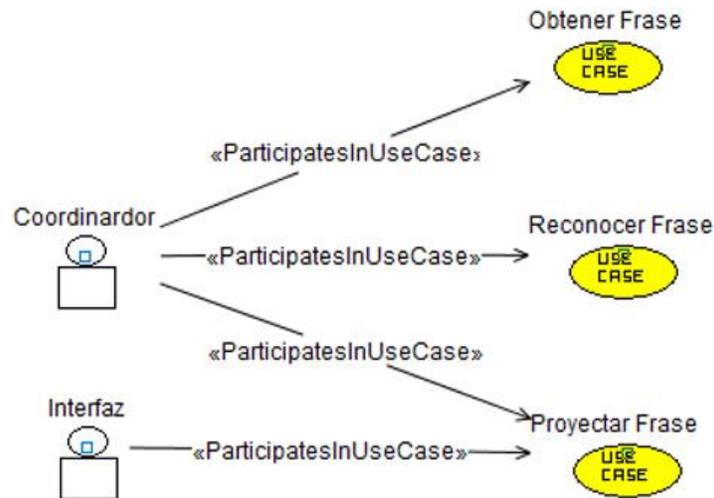


Figura 3.4 Casos de Uso

A continuación, se muestran las tablas 3.1, 3.2 y 3.3, donde se describe de forma textual los 3 casos de uso mostrados en la figura anterior.

Tabla 3.1 Caso de Uso Obtener Frase

<b>Nombre</b>	Obtener Frase
<b>Descripción</b>	Ingresa una frase al Sistema
<b>Actores</b>	Coordinador
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El Coordinador recibe una frase o palabra.</li> <li>2. El Coordinador procesa la frase utilizando CBR.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El Coordinador no recibe ninguna frase o palabra.</li> </ol>

Tabla 3.2 Caso de Uso Procesar Frase

<b>Nombre</b>	Reconocer Frase
<b>Descripción</b>	La frase o palabra ingresada es válida o no.
<b>Actores</b>	Coordinador

<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El Coordinador recibe una frase o palabra.</li> <li>2. El Coordinador envía la frase al módulo CBR.</li> <li>3. Dependiendo de la frase o palabra será la acción realizada. <ol style="list-style-type: none"> <li>3.1. Se modifica la frase hasta obtener un porcentaje de aceptación adecuado.</li> <li>3.2. Se reutiliza una misma frase previamente creada con características similares a la ingresada.</li> <li>3.3. Se crea una frase nueva, en base a las ingresadas previamente.</li> </ol> </li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El Coordinador recibe una frase o palabra.</li> <li>2. El Coordinador envía la frase al módulo CBR.</li> <li>3. Se procesa la frase.</li> <li>4. La frase no puede ser procesada por irregular o irreconocible.</li> </ol>

*Tabla 3.3 Caso de Uso Enviar Frase*

<b>Nombre</b>	Proyectar Frase
<b>Descripción</b>	La frase resultante es enviada y proyectada al usuario.
<b>Actores</b>	Coordinador e Interfaz
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El Coordinador recibe una frase o palabra.</li> <li>2. El Coordinador envía la frase al módulo CBR.</li> <li>3. Se procesa la frase.</li> <li>4. La frase resultante es enviada al módulo interfaz y la recibe el agente Interfaz.</li> </ol>

### 3.3.2. Modelo de Objetivos

A partir de los casos de uso señalados en el punto anterior, se establecieron los objetivos del sistema como se aprecia en la Fig. 3.5. A continuación, se puede apreciar el diagrama donde se muestran los objetivos mencionados, mismos que parten de un objetivo general establecido como “Procesar Frase” refiriéndose al resultado esperado. El objetivo principal se divide en: Obtener Frase, Reconocer Frase y Proyectar Frase.

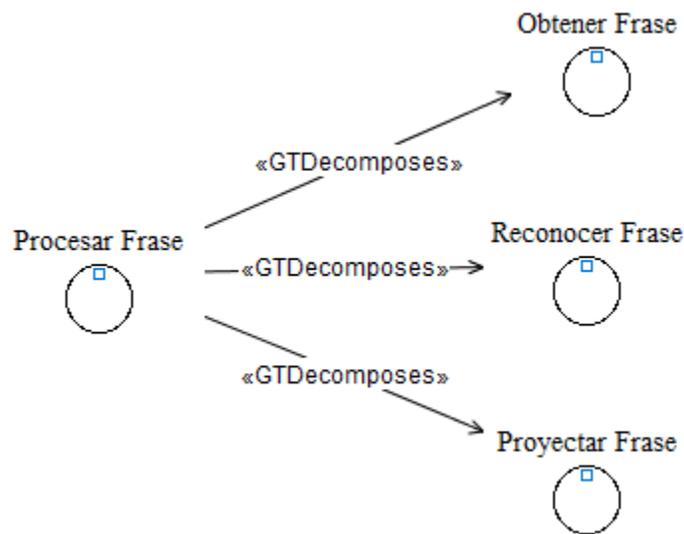
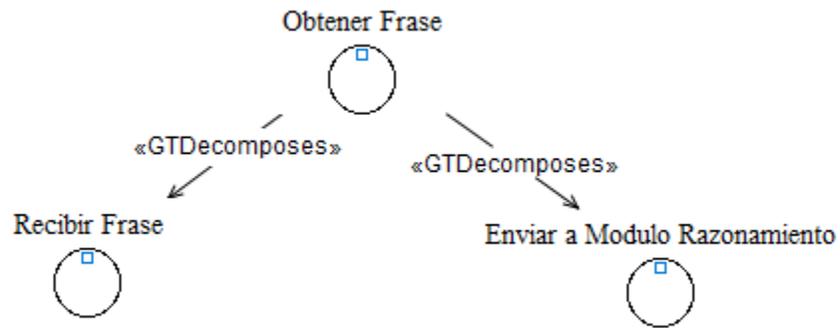


Figura 3.5 Diagrama Modelo de Objetivos

- Obtener Frase. Su función es detectar y obtener por el medio proporcionado alguna frase o palabra, para ser enviada al siguiente módulo, ver Fig. 3.6, Existe el caso donde la frase no cuenta con las características necesarias, teniendo la posibilidad de que se refiera alguna frase o palabra mal ingresada al sistema.



• *Figura 3.6 Objetivo Obtener Frase*

El primer objetivo Obtener Frase, se logra al momento de recibir una frase válida.

- Reconocer Frase. Se refiere al ingreso de la frase al sistema CBR, donde será obtenido el resultado según si necesita adaptarse, reutilizarse o crearse una nueva frase como se muestra en la Fig. 3.7.

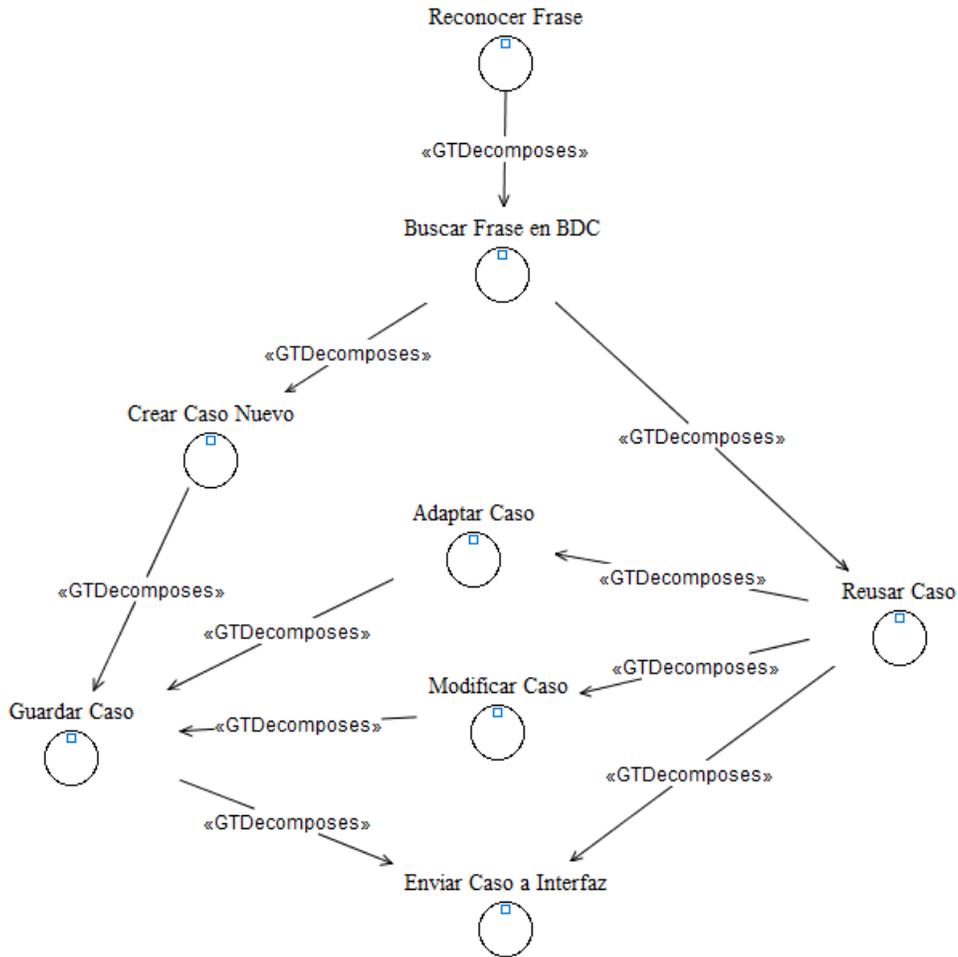
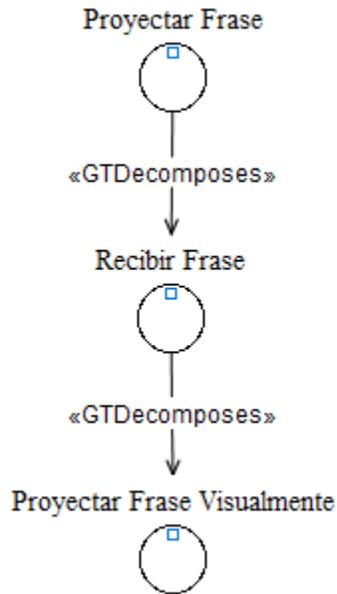


Figura 3.7 Objetivo Reconocer Frase

El segundo objetivo Reconocer Frase, se consigue al realizar una serie de tareas dependiendo de la frase ingresada, concluyendo en el envío de la misma al siguiente módulo.

- Proyectar Frase. En este último punto, el Agente Interfaz se encargará de mostrar en el entorno de la aplicación la frase resultante, para que el usuario pueda visualizarla, en este caso se sugiere el uso de una animación para la representación de la frase en LSM, en la Fig. 3.8 se muestra el proceso, pero existiendo la alternativa de poder proyectar texto o imágenes que puedan mostrar el resultado.



*Figura 3.8 Objetivo Proyectar Frase*

Dentro del objetivo Proyectar Frase, la frase elegida, deberá ser enviada y posteriormente proyectada, proyección que podrá ser realizada mediante una animación o alguna otra opción de salida, como pudiera ser una imagen, video o gif, en representación de la frase seleccionada.

### **3.3.3. Modelo de Organización**

En este punto, el modelo de organización muestra de una manera general el sistema, en el cual se representan los objetivos antes vistos y los agentes dentro del mismo. En la siguiente figura Fig. 3.9, se puede observar diagrama del modelo de organización.

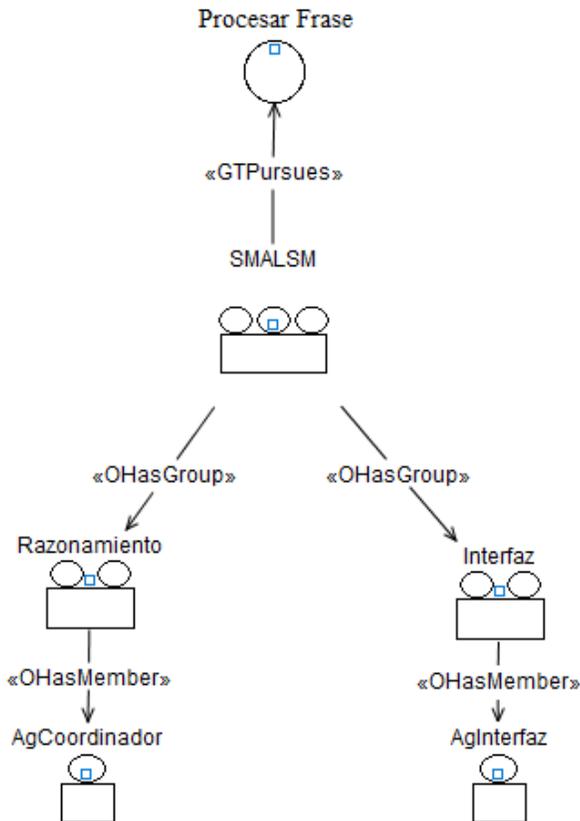


Figura 3.9 Diagrama Modelo Organización

El SMALSM (Sistema Multi-Agente de Lengua de Señas Mexicana) se compone por 2 agencias, Razonamiento e Interfaz. El primero, es el que se encarga de recibir la frase ya formada y revisar si existe el caso para ser enviado al agente Interfaz o en caso de no existir, procesar la frase para crear un nuevo caso o presentar una sugerencia que se asemeje a la información ingresada, siendo el encargado de lo descrito el agente Coordinador. Por último, el módulo de interfaz, donde el agente Interfaz se encargará de recibir la frase final y reproducir el caso en el entorno de la aplicación.

### 3.3.4. Modelo de Agentes

En el modelo de agentes, como se aprecia en la Fig. 3.10, se muestran las tareas que realizara cada uno de ellos, en este caso para el SMALSM se requerirán 2 agentes, los cuales son: Agente Coordinador y Agente Interfaz, los cuales corresponden respectivamente a los módulos de Razonamiento y al módulo Interfaz. En la siguiente figura Fig. 3.10, se muestra el modelo de agentes.

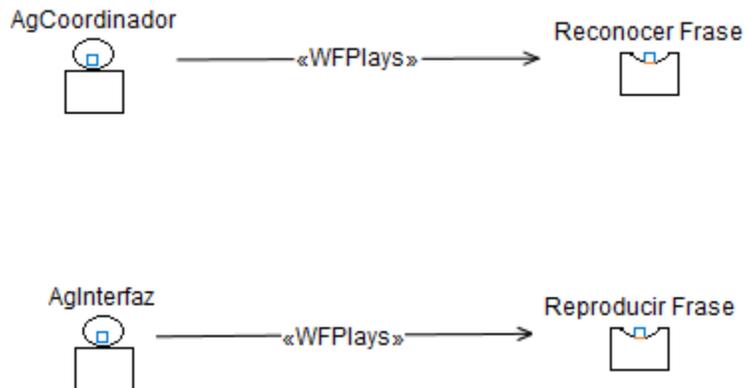


Figura 3.10 Diagrama Modelo de Agentes

Las tareas asignadas a cada agente que se muestran en la Fig. 3.11, son:

- Reconocer Frase. Se deberá decidir si la frase es adecuada para ser utilizada o no, recuperarla, almacenarla y enviarla.
  1. Evaluar Frase, por lo tanto, es aceptada o no, para su uso,
  2. Recuperar Frase, la frase se admite y en caso de existir en la base de creencias se reutiliza o se modifica como una nueva creencia.
- Almacenar Frase. Esta tarea se encarga de almacenar la nueva frase o la frase modificada, en la base de creencias.
- Enviar Frase para Proyectarla. La frase es enviada con el Agente Interfaz, para proyectarla.

Las tareas del Agente Interfaz, se detallan tareas a continuación.

- Recibir Frase. Recibe la frase final creada o reutilizada.
- Proyectar Visualmente Frase. La frase resultante es proyectada al usuario mediante una animación, imagen, video o gif.

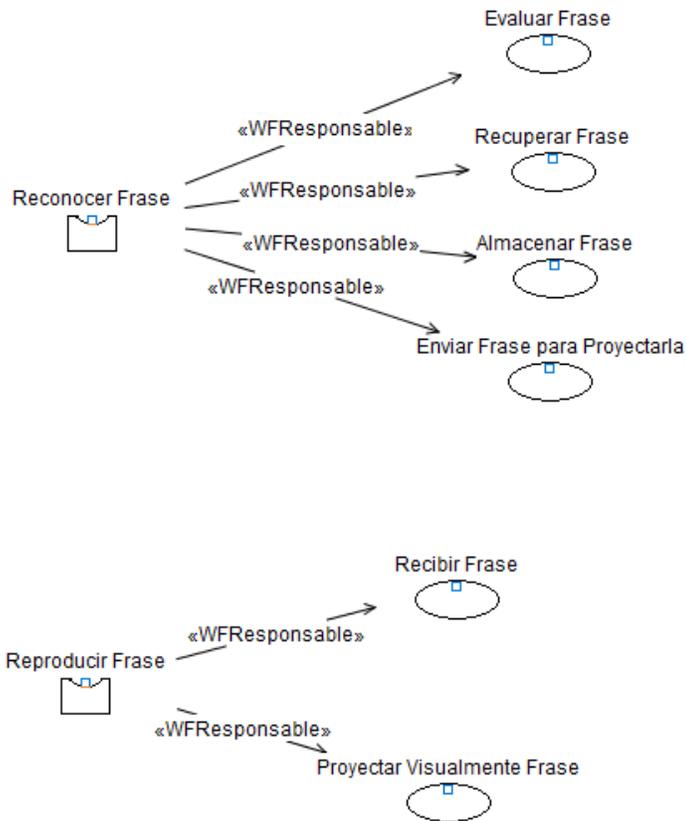


Figura 3.11 Diagrama de Tareas

### 3.3.5 Modelo de Interacciones

En este modelo se describe el comportamiento y comunicación que tiene cada agente dentro de su respectivo módulo que componen el sistema propuesto como se muestra en la Fig. 12.



Figura 3.12 Diagrama de Interacciones

En este diagrama se puede observar como el inicio de la interacción entre agentes, es el reconocimiento de la frase, misma que será reproducida una vez termine todo el proceso.

## **Capítulo 4: Implementación**

En este capítulo se presenta el desarrollo del sistema en base a los modelos mostrados en el capítulo anterior, describiendo así el funcionamiento del razonamiento basado en casos y del uso que le dan al mismo los agentes involucrados y la comunicación entre estos.

La implementación del sistema se llevó a cabo con una simulación utilizando la plataforma NetBeans, donde se muestran las tablas de la base de creencias, creadas utilizando mySql en XAMPP y con el IDE de JADE, para la creación y programación de los agentes.

### **4.1 Modelo de Interacciones**

Para la creación del Sistema CBR, se consideró el uso de 4 tablas, las cuales contendrían palabras y frases asociadas entre sí, las cuales servirían para iniciar con el proceso que permitiría obtener resultados satisfactorios en la traducción.

### **4.2 Creando la Base de Creencias**

Para iniciar con la construcción del sistema CBR, primero se creó la base de creencias necesaria para su funcionamiento y dentro de esta, las tablas necesarias.

Utilizando XAMPP, se creó la base de datos llamada “frases” y dentro de ella las 4 tablas llamadas “palabras”, “palabras2”, “frase” y “frase2”, en la Fig. 4.4 se muestran las tablas, mismas que como dicen su nombre almacenarán palabras y frases, que serán utilizadas para el funcionamiento del sistema. Más adelante se dará mayor detalle al funcionamiento de las mismas.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
frase	Examinar Estructura Buscar Insertar Vaciar Eliminar	109	InnoDB	latin1_swedish_ci	16 KB	-
frase2	Examinar Estructura Buscar Insertar Vaciar Eliminar	45	InnoDB	latin1_swedish_ci	16 KB	-
palabras	Examinar Estructura Buscar Insertar Vaciar Eliminar	142	InnoDB	latin1_swedish_ci	16 KB	-
palabras2	Examinar Estructura Buscar Insertar Vaciar Eliminar	126	InnoDB	latin1_swedish_ci	32 KB	-
4 tablas	Número de filas	422	InnoDB	latin1_swedish_ci	80 KB	0 B

Figura 4.1 Tablas de la Base de Creencias

## 4.2.1 Estableciendo la Conexión

Una vez teniendo la Base de Creencias creadas, se establece como será la conexión con esta. La programación fue realizada en el lenguaje de programación Java, en la plataforma NetBeans, incluyendo también la librería necesaria para iniciar el servicio de MySQL. En la Fig. 4.2 se muestra el código en Java para la conexión y en la Fig. 4.3 se muestra la librería usada para la conexión.

```

package automata.palabras;

import java.sql.Connection;
import java.sql.DriverManager;

public class Conexion {

    public Connection conexion(){
        Connection cn = null;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/frases","root","");
        } catch (Exception e) {
        }
        return cn;
    }
}

```

Figura 4.2 Conexión a Base de Creencias

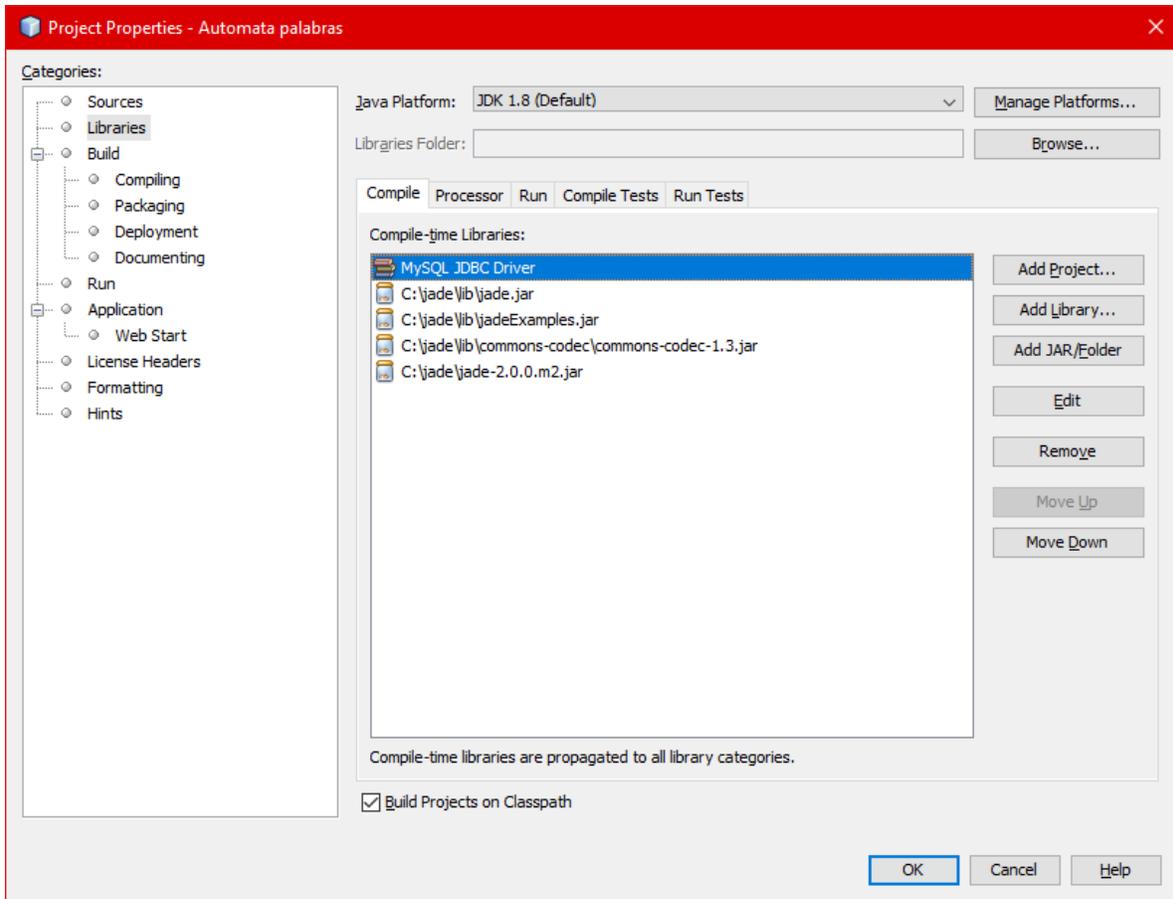


Figura 4.3 Librería de MySQL

### 4.3. Manejo de Frases

Para poder iniciar con el reconocimiento de frases del sistema CBR, se implementó una entrada sencilla para las mismas, siendo esta, proporcionada por el usuario a la consola para fines de la simulación, en la Fig. 4.4 se muestra el código para este proceso.

```
package automata.palabras;

import java.util.Scanner;

public class PedirFrase {
    String frase;
    Scanner sc = new Scanner(System.in);

    public void setFrase(){
        System.out.println("Introduzca la frase: ");
        frase = sc.nextLine();
    }

    public String getFrase(){
        return frase;
    }
}
```

*Figura 4.4 Ingreso de Frase*

### 4.3.1. Separación de Palabras

Una vez contando con la frase o palabra ingresada, esta es dividida y vuelta a unir, y así identificar los espacios vacíos y la cantidad de palabras utilizadas, en la Fig. 4.5 se muestra el código para este proceso.

```

package automata.palabras;

public class SepararFrase {
    String[] palabra = new String[1000];
    String frase = "", pf;
    int j = 1;

    SepararFrase(String palabraRecibida) {
        pf = palabraRecibida;
    }

    void SepararPalabras() {
        char caracter;
        int estado = 0, columna, valorMatriz;
        palabra[j] = "";

        int[][] matriz = {
            //1    eb    oc
            /*0*/    {1, 0, 200},
            /*1*/    {1, 100, 200}
        };

        for (int i = 0; i < pf.length(); i++) {
            caracter = pf.charAt(i);

            if (Character.isLetter(caracter)) {
                columna = 0;
            } else if (caracter == 32) {
                columna = 1;
            } else {
                columna = 2;
            }
            valorMatriz = matriz[estado][columna];

            if (valorMatriz <=1) {
                estado = valorMatriz;
                frase = frase + caracter;
                palabra[j] = palabra[j] + caracter;
            } else if (valorMatriz == 200) {
                System.out.println("Invalido");
            } else {
                System.out.println("Palabra " + j + " = " + palabra[j]);
                j++;
                frase = frase + " ";
                palabra[j] = "";
            }
        }
        System.out.println("Palabra " + j + " = " + palabra[j]);
        System.out.println("\nEsta es la Frase: " + frase + "\n");
    }
}

```

Figura 4.5-1 Separación de Palabras

```

public String[] getPalabra() {
    return palabra;
}

public String getFrase() {
    return frase;
}

public int getJ() {
    return j;
}
}

```

*Figura 4.5-2 Separación de Palabras*

### **4.3.2. Primera Tabla Palabras**

Para iniciar con la primera tabla “palabras”, esta deberá contar con una serie de palabras previamente cargadas, para así trabajar con ellas. Primero, de la frase que fue dividida en el punto anterior, es analizada y comparada palabra por palabra con las ya existentes en la primera tabla “palabras”, que en caso de estar presentes todas, no hay cambios, en caso contrario la palabra no encontrada dentro de la lista, es añadida como una nueva, siendo esta reconocida la siguiente vez que se ingrese una nueva frase con dicha palabra en ella. Las palabras previamente añadidas a la primera tabla, están asociadas a su seña correspondiente, al actualizarse la tabla con nuevas palabras, estas pasarían a ser deletreadas y en caso de ser posible asociarla con la seña que la describa o asociada como sinónimo de alguna palabra ya existente más adelante, en la Fig. 4.6 se muestra el código para este proceso.

```

package automata.palabras;

import static automata.palabras.AutomataPalabras.palabra;
import java.sql.SQLException;
import java.sql.Statement;

public class CompararPalabras {
    String[] palabral = new String [100000];
    DatosConexion da = new DatosConexion();
    PedirFrase pf = new PedirFrase();
    SepararFrase sf = new SepararFrase(palabra);

    void compararPalabras()throws ClassNotFoundException, SQLException{
        da.setProbarconexion();
        pf.setFrase();
        palabra = pf.getFrase();
        sf = new SepararFrase(palabra);
        sf.SepararPalabras();
        Statement instruccion = da.cn.createStatement();

        for (int i = 1; i <= sf.getJ(); i++) {
            for (int j = 1; j < da.j; j++) {
                palabral[i] = sf.getPalabra()[i];

                if (!palabral[i].equals(da.getPalabra1()[j]) && j==da.j-1 ) {
                    instruccion.execute("INSERT INTO `palabras` (id_palabra, "
                        + "palabra) VALUES (NULL, '"+palabral[i]+'');");
                    da.setProbarconexion();
                } else if(palabral[i].equals(da.getPalabra1()[j])){
                    j=da.j-1;
                }
            }
        }
    }
}

```

Figura 4.6 Comparación Palabras Primera Tabla

### 4.3.3. Segunda Tabla Palabras

La segunda tabla, “palabras2”, similar a la primera tabla contiene una lista de palabras, pero a diferencia de la primera esta tabla se crea desde cero con las palabras que conforman las frases, en esta también se toma en cuenta la palabra inicial de cada frase y en base a ella se agregan las palabras, siendo el caso de no existir alguna palabra actualizar la tabla y si esta ya se encuentra, aumentar un

contador, indicando así la cantidad de veces que se ha utilizado la palabra, después de la palabra inicial, en la Fig. 4.7 se muestra el código para este proceso.

```

package automata.palabras;

import java.sql.SQLException;
import java.sql.Statement;

public class CompararPalabras2 extends CompararPalabras{
    int identificador = 0, contador = 1;

    void compararPalabras2 () throws SQLException {
        da.setProbarconexion();
        Statement instruccion = da.cn.createStatement();
        for (int i = 1; i <= sf.getJ(); i++) {
            for (int k = 0; k < da.k; k++) {
                for (int j = 1; j < da.j; j++) {
                    if (super.palabral[i].equals(da.getPalabral()[j]) && i==1) {
                        identificador = da.getId()[j];
                    }
                }

                if (super.palabral[i].equals(da.getPalabra2()[k]) && identificador==da.getIdentificador()[k]) {
                    contador = da.getContador()[k] + 1;
                    instruccion.execute("UPDATE `palabras2` SET contador = '"+contador+"' WHERE id_palabra2 in ('"+k+"')");
                    k=da.k;
                } else if (k==da.k-1 && (!super.palabral[i].equals(da.getPalabra2()[k]) ||
                    super.palabral[i].equals(da.getPalabra2()[k]) && identificador!=da.getIdentificador()[k])) {
                    contador = 1;
                    System.out.println(super.palabral[i]+": no se encuentra en la segunda tabla, agregando...");
                    instruccion.execute("insert into `palabras2` (id_palabra2, palabra2, identificador, contador) "
                        + "values (NULL, '"+super.palabral[i]+"', '"+identificador+"','"+contador+" )");
                    k=da.k;
                }
            }
        }
        da.setProbarconexion();
    }
}

```

Figura 4.7 Comparación Palabras Segunda Tabla

Por ejemplo, si se ingresa la frase “Ayer tomé agua”, a la segunda tabla se añadirían las 3 palabras, tomando de referencia la palabra inicial, en este caso “Ayer”, siendo así 3 palabras nuevas añadidas a la tabla. Si consecutivamente se ingresa la frase “Hoy tomé agua”, las palabras “tomé” y “agua”, ya existen, pero la palabra inicial es diferente, por lo tanto, se añaden como nuevas palabras referenciadas a la palabra inicial “Hoy” y por último en caso de ingresar una frase con una palabra inicial anteriormente utilizada, como “Ayer tomé soda” u “Hoy tomé soda”, la palabra “tomé” en ambos casos aumenta el contador indicando que

se ha vuelto a utilizar después de la misma palabra por segunda ocasión, a diferencia de la palabra “soda” que es la primera vez que es utilizada.

#### 4.3.4. Primera y Segunda Tabla Frases

La siguiente tabla “frase”, hace uso de la segunda tabla de palabras vista en el punto de anterior y como dice su nombre, en esta están incluidas las frases ya formadas en el sistema y de manera similar a como la tabla “palabras2” se relaciona a la tabla “palabras”, las frases de la tabla “frase” están relacionadas a las palabras de la tabla “palabras2” y se señala también la cantidad que estas han sido utilizadas.

En relación con la primera tabla de frases, se creó una segunda tabla “frase2”, en la cual son agregadas las nuevas frases ingresadas al sistema, que sean sinónimos, de algunas de las frases almacenadas previamente en la primera tabla de frases, en la Fig. 4.8 se muestra el código para este proceso.

```
package automata.palabras;

import static automata.palabras.AutomataPalabras.frase;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class Frases {
    int[] identificador=new int[10000];
    int contador = 0;
    String [] f = new String[10000];
    int[] cantidad = new int[1000], id = new int[1000];
    int l=1;
    String Resultado;

    void frases () throws SQLException, ClassNotFoundException{
        CompararPalabras2 cp2 = new CompararPalabras2();
        cp2.compararPalabras();
        cp2.compararPalabras2();
        int[] identificador=new int[10000];
        int contador = 0;
        Statement instruccion = cp2.da.cn.createStatement();
        frase=cp2.pf.getFrase();
        Scanner valorId = new Scanner (System.in);
        String valor = "";
        int g= 0;

        for (int i = 1; i <= cp2.sf.getJ(); i++) {
            for (int j = 0; j < cp2.da.l; j++) {
                for (int k = 1; k <= cp2.da.k-1; k++) {
                    if (cp2.da.getPalabra2()[k].equals(cp2.sf.getPalabra()[i]) && cp2.identificador == cp2.da.getIdentificador()[k]) {
                        Resultado = cp2.da.getSeña()[j];
                        identificador[i] = cp2.da.getId2()[k];
                        k=cp2.da.k-1;
                    }
                }
            }

            if (frase.equals(cp2.da.getFrase()[j]) && identificador[i]==cp2.da.getIdentificador2()[j]) {
                Resultado = cp2.da.getSeña()[j];
            }
        }
    }
}
```

Figura 4.8-1 Creación de Tablas “frase” y “frase2”

```

        contador = cp2.da.getContador2()[j]+1;
        instruccion.execute("UPDATE `frase` SET contador = '"+contador+"' WHERE id_frase in ('"+j+"')");

        for (int k = 1; k < cp2.da.l; k++) {
            while (identificador[i]==cp2.da.getIdentificador2()[k]) {
                contador = cp2.da.getContador2()[k]+1;
                instruccion.execute("UPDATE `frase` SET contador = '"+contador+"' WHERE id_frase in ('"+k+"')");
                k++;
            }
        }
    }

    else if (frase.equals(cp2.da.getFrase2()[j]) ) {

        System.out.println("Existe la frase en la segunda tabla seña aqui " + j);
        identificador[j] = cp2.da.getIdentificador3()[j];
        Resultado = cp2.da.getSeña()[identificador[j]];
        contador = cp2.da.getContador2()[identificador[j]+1];

        instruccion.execute("UPDATE `frase` SET contador = '"+contador+"' WHERE id_frase in ('"+identificador[j]+"')");
        j=cp2.da.l;

    } else if (frase.equals(cp2.da.getFrase()[j]) && identificador[i]!=cp2.da.getIdentificador2()[j] && j==cp2.da.l-1 ||
        !frase.equals(cp2.da.getFrase()[j]) && j==cp2.da.l-1) {
        if (g==0) {
            Resultado = cp2.da.getSeña()[j];
            ResultSet rs = instruccion.executeQuery("SELECT * FROM frase INNER JOIN palabras2 ON id_palabra2 = frase.identificador "
                + "WHERE frase.identificador = '"+identificador[i]+'\" ORDER BY frase.contador DESC LIMIT 10");
            while (rs.next())
            {
                id[l] = rs.getInt(1);
                f[l] = rs.getString(2);
                cantidad[l] = rs.getInt(4);
                System.out.println(id[l]+" : "+f[l]+" "+cantidad[l]);
                l++;
            }
            if (rs.next()==false) {
                Resultado = cp2.da.getSeña()[j];
                System.out.println("La frase "+frase+" no se encuentra, DA ENTER para agregarla");
            } else{
                System.out.println("La frase "+frase+" no se encuentra, DA ENTER para agregarla o "
                    + "selecciona un id de las frases dadas para relacionar la la frase nueva con la que ya existe");
            }
        }
        valor = valorId.nextLine();
    }
    for (int k = 1; k < cp2.da.l; k++) {
        while (identificador[i]==cp2.da.getIdentificador2()[k]) {
            Resultado = cp2.da.getSeña()[j];
            contador = cp2.da.getContador2()[k]+1;
            instruccion.execute("UPDATE `frase` SET contador = '"+contador+"' WHERE id_frase in ('"+k+"')");
            k++;
        }
        if (!valor.equals("")) {
            Resultado = cp2.da.getSeña()[j];
            instruccion.execute("INSERT INTO `frase2` (frase2, identificador) VALUES ('"+frase+"', '"+valor+"')");
        } else {
            Resultado = cp2.da.getSeña()[j];
            contador=1;
            instruccion.execute("INSERT INTO `frase` (id_frase, frase, identificador, contador) VALUES (NULL, '"+frase+"', '"
                +identificador[i]+'','"+contador+"')");
            g=1;
        }
    }
}

}

public String[] getF(){
    return F;
}

public int[] getCantidad(){
    return cantidad;
}

public int getL(){
    return l;
}

public String getResultado(){
    return Resultado;
}
}

```

Figura 4.8-2 Creación de Tablas "frase" y "frase2"

Por ejemplo, iniciando con la primera tabla de frases “frase”, al momento de ingresar la información, en caso de ser la primera vez, se creará como un nuevo registro, en el cual se podrá observar la frase completa ingresada, los “id” de las palabras de la tabla “palabras2” a las cuales se encuentra relacionada y la cantidad de veces que la frase ha sido ingresada al sistema, en este caso 1 vez, siendo el caso de la frase “ayer tome agua”, esta sería agregada como 3 registros diferentes, 1 vez por cada palabra dentro de la frase, a las estaría relacionada respectivamente, y así de esta manera llevar una estadística de uso de frases, que permitiría obtener resultados más precisos una vez que el sistema cuente con gran variedad de registros.

Continuando con la segunda tabla de frases “frase2”, teniendo esta un funcionamiento más simple que la primera, se encargaría de almacenar frases, que sean consideradas sinónimos de frases existentes en la primera tabla, como ejemplo al ingresar la frase “ayer bebí agua”, se puede decir que es sinónimo del ejemplo utilizado anteriormente “ayer tomé agua” y siendo este el caso, las frases serían relacionadas entre sí, añadiendo a la segunda de estas en la segunda tabla de frases, decisión que sería tomada en este caso por el agente encargado del CBR, el agente Coordinador, mismo que será descrito más adelante, pero que gracias a ello, sin importar cuál de estas dos frases sea la que sea nuevamente ingresada, el resultado pueda ser el mismo.

#### **4.4. Agentes Coordinador e Interfaz**

Para este proyecto, fue necesario la creación de dos agentes con funciones distintas, utilizando la herramienta JADE, como se muestra en la Fig. 4.9. Como se mencionó en el capítulo anterior, uno de los agentes (el Agente Coordinador) es el encargado de recibir la frase y se encarga de tomar la decisión de si aceptarla o no, dependiendo de las características de la misma, una vez que la frase es aceptada este sería el encargado de procesarla utilizando CBR descrito anteriormente, obteniendo así un resultado, el cual sería enviado por este mismo

agente al segundo (el Agente Interfaz), para que este último se encargara de su proyección en la interfaz del usuario.

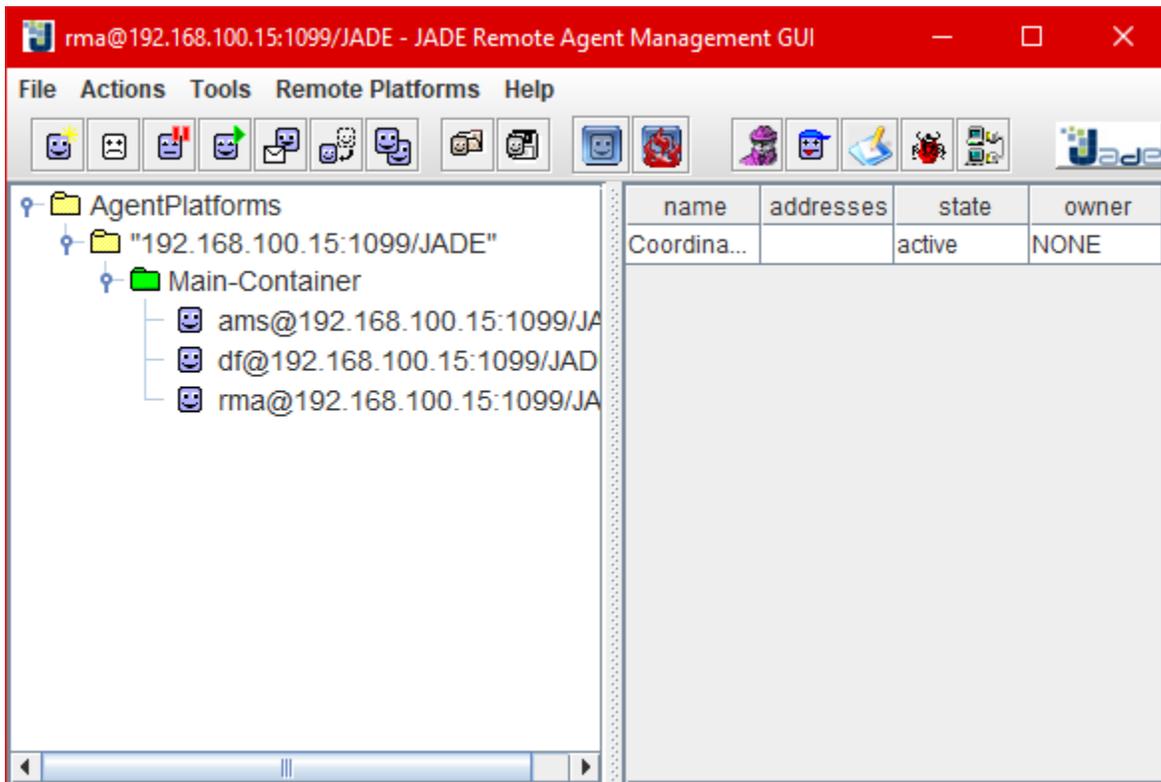


Figura 4.9 Interfaz del IDE JADE

#### 4.4.1. Creando al Agente Coordinador y al Agente Interfaz

A cada uno de los dos agentes, se le creó su propia clase, en las figuras Fig. 4.10 y 4.11 se muestra el código correspondiente, las cuales les indican los comportamientos que estos tendrán.

En este caso, ambos agentes utilizarán un mismo comportamiento, denominado CyclicBehaviour, el cual se encarga de mantenerse repitiendo la misma acción,

El agente Coordinador consta de dos principales acciones, las cuales son mantener en ejecución el sistema y enviar el resultado al agente Interfaz, mientras que el mencionado agente Interfaz de igual manera consta de dos acciones, siendo estas la de recibir el mensaje del agente Coordinador y de mostrarlo en la interfaz de usuario.

```

package automata.palabras;
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import java.sql.SQLException;

public class Coordinador extends Agent{

    @Override
    protected void setup(){

        addBehaviour(new CyclicBehaviour(){

            @Override
            public void action(){
                Frases frases = new Frases();

                try {
                    frases.frases();
                } catch (SQLException ex) {
                } catch (ClassNotFoundException ex) {
                }

                String Resultado = frases.getResultado();
                //enviar mensaje
                ACLMessage msg=new ACLMessage (ACLMessage.INFORM);
                msg.setContent (Resultado);
                msg.addReceiver(new AID("interfaz", AID.ISLOCALNAME));
                send(msg);
            }
        });
    }
}

```

Figura. 4.10 Clase del Agente Coordinador

```

package automata.palabras;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import java.awt.event.ActionEvent;
import javax.swing.ImageIcon;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.Timer;

public class Interfaz extends Agent{
    @Override
    protected void setup(){
        addBehaviour(new CyclicBehaviour(){
            @Override
            public void action(){
                //recibir mensaje
                ACLMessage msg = receive();
                if (msg!=null) {
                    ImageIcon dab = new ImageIcon(msg.getContent());
                    JDialog dialog = new JDialog();
                    dialog.setUndecorated(true);
                    JLabel label = new JLabel( dab);
                    dialog.add( label );
                    dialog.pack();
                    dialog.setVisible(true);

                    Timer timer = new Timer(2000, (ActionEvent e) -> {
                        dialog.setVisible(false);
                        dialog.dispose();
                    });
                    timer.setRepeats(false);
                    timer.start();
                    dialog.setVisible(true);
                }else block();
            }
        });
    }
}

```

Figura 4.11 Clase del Agente Interfaz

Aunque estas son las clases que definirán el comportamiento inicial de los agentes, estos mismos aún no han sido creados y a diferencia de los objetos,

estos se crean de distinta forma, la primera es utilizando la interfaz mostrada en la Figura 4.8, claro, esto dependiendo de la utilidad de los agentes o si algún o de ellos se necesita o puede crearse más tarde, pero para este caso, los agentes se crearán una vez inicie el programa, para ello habrá que dirigirse a las propiedades del proyecto y especificar cuantos y cuales agentes necesitamos y como iniciará el programa como se aprecia en la Fig. 4.12.

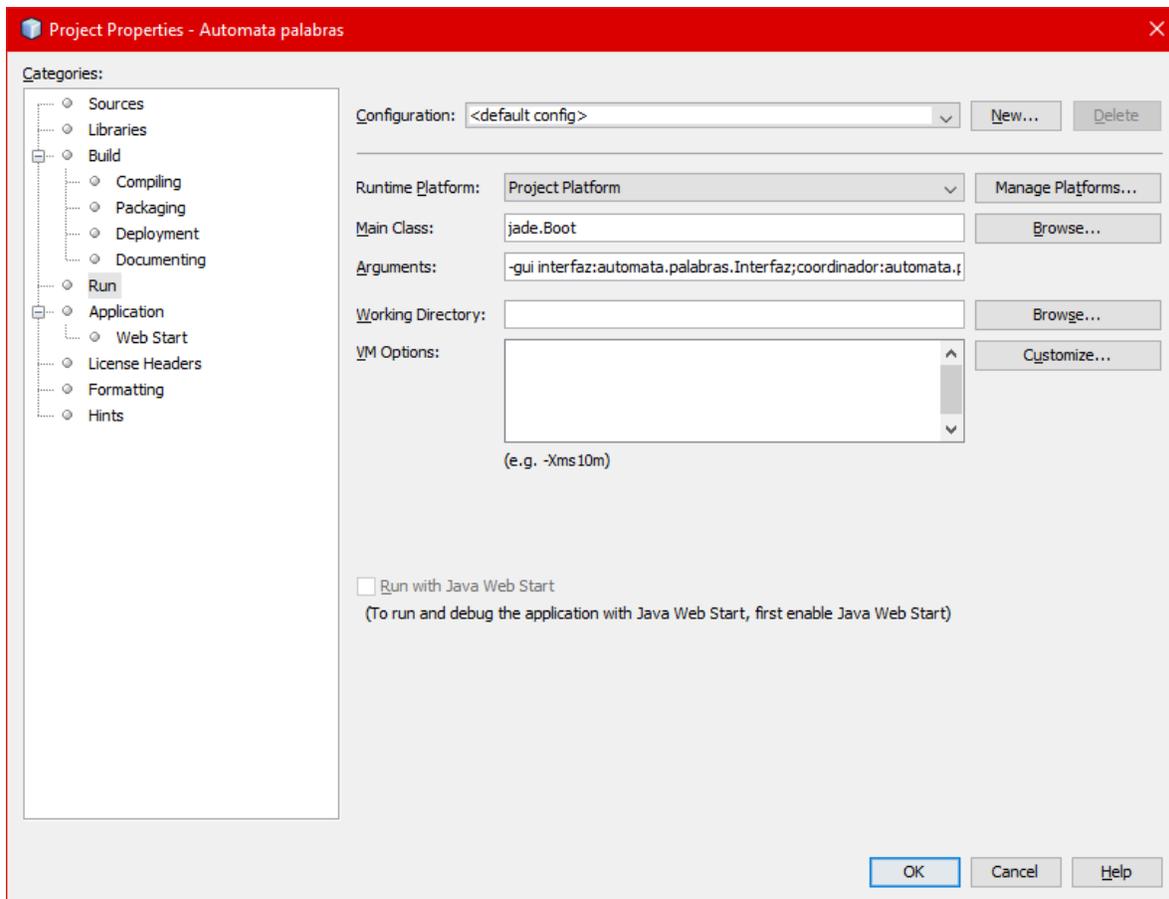


Figura 4.12 Propiedades del Proyecto

Como se muestra en la anterior figura, la clase principal es jade.Boot, esto ejecutará una vez iniciada la simulación, la interfaz de JADE y en la siguiente casilla “Argumentos”, es donde se crean los agentes seguido de la clase a la que corresponden de la siguiente manera “nombredelagente:nombredelpaquetedelproyecto.nombredelaclasedelagente”, y

para agregar un segundo o más agentes, es necesario separarlos con “;”, los agentes también pueden ser creados desde la misma IDE de JADE, pero en este caso es más de mayor facilidad que los agentes se creen al momento de ejecutar el sistema.

De igual manera deberán agregarse las librerías necesarias al proyecto, las cuales contienen los datos para la ejecución de los agentes, como se aprecia en la Fig. 4.13.

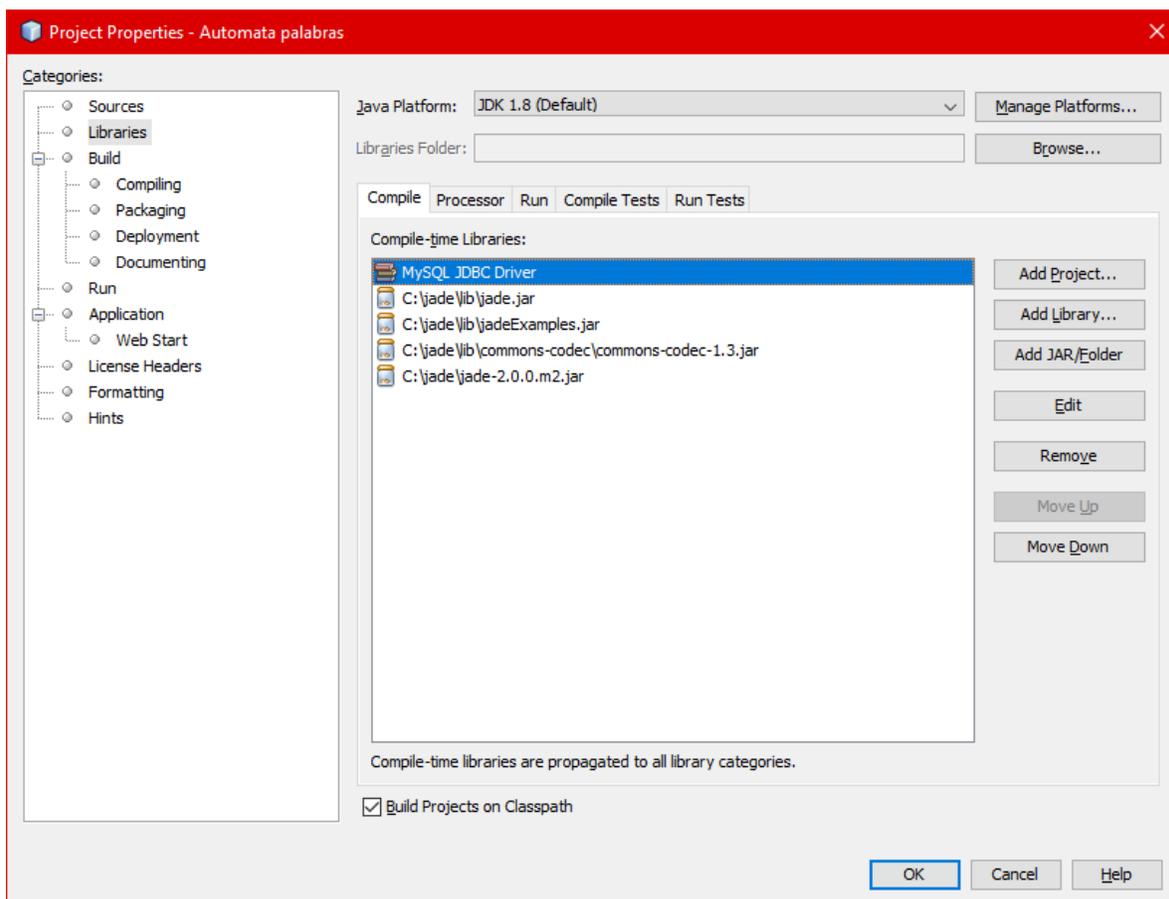


Figura 4.13 Librerías Agregadas a las Propiedades del Proyecto

#### 4.4.2. Grafo de flujo del Sistema

Para representar la secuencia que sigue el sistema, descrita en los puntos anteriores y según las condiciones cumplidas, en la siguiente Fig. 4.14, se muestra el grafo de flujo resultante.

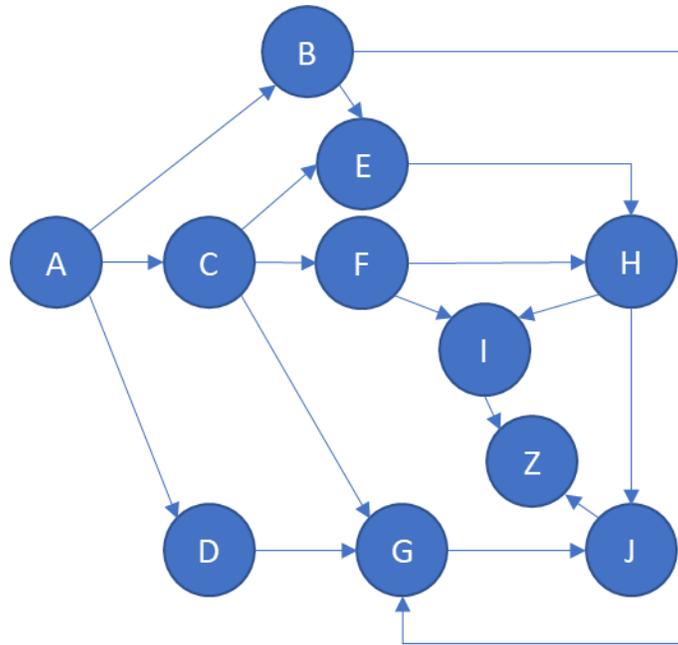


Figura 4.14 Grafo del Sistema

La secuencia mostrada en el diagrama anterior, se compone de 11 estados posibles, clasificados desde la 'A' a la 'J', siendo el estado final la 'Z', como se describe en la Tabla 4.1.

Tabla 4.1 Descripción estados del sistema

Estado	Descripción
A	Ingreso de una frase o palabra
B	Algunas palabras se encuentran registradas
C	Todas las palabras se encuentran registradas
D	Ni una palabra se encuentra registrada
E	Sólo la Palabra Inicial se reconoce
F	Las palabras son reconocidas
G	Las palabras no se reconocen
H	Se encuentra frase similar
I	La frase se utiliza
J	Se añade como nueva frase
Z	El Agente Coordinador envía la frase al Agente Interfaz y este la muestra

## **Capítulo 5: Análisis de resultados**

En el presente capítulo se hablará, de las pruebas y de los resultados obtenidos y los objetivos logrados dentro de este proyecto, mencionando logros importantes y observaciones necesarias para la continuación del mismo. Se verán también posibles trabajos futuros relacionados a este y posibles implementaciones a proyectos anteriormente realizados, que pudieran mejorar considerablemente la visión que se tiene para este proyecto, convirtiéndolo en una excelente oportunidad para trabajos y desarrollos futuros.

### **5.1. Pruebas del Sistema**

Para la realización de pruebas del sistema, se consideran solo palabras conocidas del español, evitando en lo posible regionalismos, anglicismos y barbarismos, de manera que se utilice un español lo más neutral posible, aplicándolo en oraciones cortas y sencillas, suponiendo que las mismas, ya se encuentren en gramática de Lengua de Señas Mexicana.

Las pruebas a realizarse son de tipo funcional, refiriéndose a la funcionalidad del producto de software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones específicas [25].

El tipo de técnica con la que se realizarán estas pruebas, son de tipo dinámica, lo cual indica que serán realizadas ejecutando la aplicación y así probarlo con una visión interna, cerciorándose que todos los módulos encajen y funcionen unos con otros, llamándose a estas pruebas “Pruebas de caja blanca”. Estas pruebas garantizan que todas las rutas del código se revisan al menos una vez, se revisan las condiciones lógicas y las estructuras de datos. Para estas pruebas se empleará el método de ruta básica, propuesto por Tom McCabe. Para representar estas pruebas se utilizará a lo que se conoce como grafo de flujo, donde se mostrarán los diferentes caminos que puede tomar el sistema durante su ejecución [26]. También existen las pruebas de caja negra, realizadas desde una

visión externa, intentando encontrar errores en las funciones, pero para este caso se realizarán las pruebas de caja blanca.

### **5.1.1. Pruebas de Ruta Básica**

A continuación, en la Figura 5.1, utilizando el método de ruta básica, se muestran diferentes diagramas de grafo, donde se representan los diferentes caminos posibles dentro del sistema, correspondiendo al inciso 'a' el Caso 1, al 'b' al Caso 2 y así sucesivamente hasta el caso número 10, siendo estas el número de posibles caminos a seguir dentro del sistema.

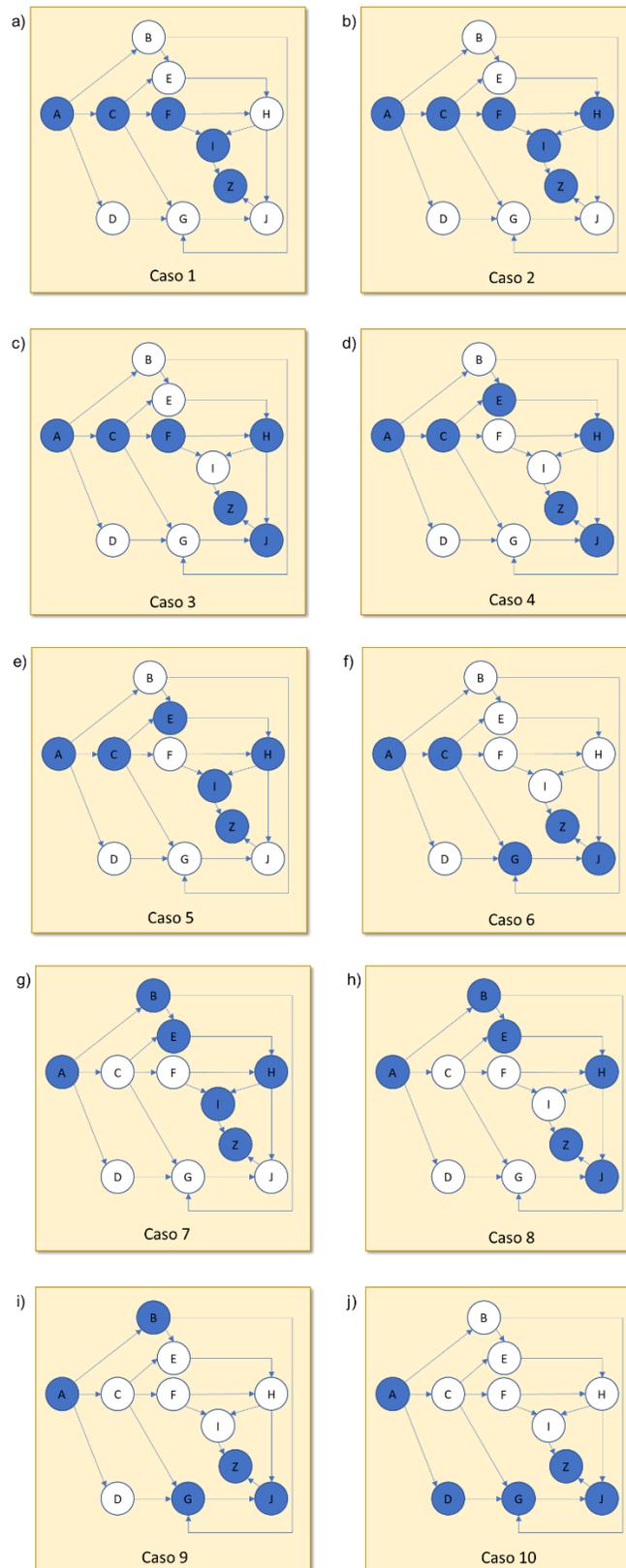


Figura 5.1 Casos de Prueba Obtenidos. a) Caso 1. b) Caso 2. c) Caso 3. d) Caso 4. e) Caso 5. f) Caso 6. g) Caso 7. h) Caso 8. i) Caso 9. j) Caso 10.

Todos los casos inician con el primer estado nombrado 'A', el cual se refiere al ingreso de información (frase/palabra) al sistema y finalizan en el estado 'Z', el cual se refiere a la representación visual que proporciona el agente Interfaz, después de recibir la frase resultante. Los estados fueron nombrados de la 'A' a la 'J', exceptuando el estado final, mencionado anteriormente.

Para el análisis de la secuencia de ruta básica anteriores, se utilizaron mensajes en consola, indicando en estos que los casos se cumplen en las condiciones especificadas en un 100%.

### 5.1.2. Interpretación de los resultados

Cada una de las rutas puede llegar a dos distintos resultados previos, los cuales son o existe la frase o no existe, aunque estos dos resultados son interpretados igualmente por el agente Interfaz, mostrando la seña de forma gráfica, el cual es el resultado final, mientras que cada caso inicia de la misma forma, con el ingreso de la frase o palabra al sistema, ejecutado por el agente coordinador, siendo, en caso de una frase, dividida en las palabras que la conforman para indicar si ya se encuentran registradas o no en el sistema. A continuación, se observa en la Tabla 5.1, la interpretación de los casos mostrados en el punto anterior.

*Tabla 5.1 Interpretación de rutas de prueba*

<b>Número de caso</b>	<b>Descripción</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
Caso 1	A: Ingresa Frase/Palabra. C: La/s Palabras están Registradas. F: Las Palabras se Reconocen. I: La Frase Encontrada puede Utilizarse.	Reconocimiento de todas las palabras y la frase.	Reconocimiento de todas las palabras y la frase.

	Z: El Agente Interfaz Muestra la Señal.		
Caso 2	A: Ingresa Frase/Palabra. C: La/s Palabras están Registradas. F: Las Palabras se Reconocen. H: Se Encuentra Frase Similar. I: La Frase Encontrada puede Utilizarse. Z: El Agente Interfaz Muestra la Señal.	Reconocimiento de todas las palabras, frase similar y asociación con frase similar.	Reconocimiento de todas las palabras, frase similar y asociación con frase similar.
Caso 3	A: Ingresa Frase/Palabra. C: La/s Palabras están Registradas. F: Las Palabras se Reconocen. H: Se Encuentra Frase Similar. J: Se Añade Nueva Frase. Z: El Agente Interfaz Muestra la Señal.	Reconocimiento de todas las palabras, frase similar, pero se añade como una nueva.	Reconocimiento de todas las palabras, frase similar, pero se añade como una nueva.
Caso 4	A: Ingresa Frase/Palabra. C: La/s Palabras están Registradas. E: Sólo la Palabra Inicial se Reconoce. H: Se Encuentra Frase Similar. J: Se Añade Nueva Frase. Z: El Agente Interfaz Muestra la Señal.	Reconocimiento de algunas de las palabras, reconocimiento de frase, pero se añade como frase nueva.	Reconocimiento de algunas de las palabras, reconocimiento de frase, pero se añade como frase nueva.
Caso 5	A: Ingresa Frase/Palabra. C: La/s Palabras están Registradas.	Reconocimiento de algunas de las palabras,	Reconocimiento de algunas de las palabras,

	<p>E: Solo la Palabra Inicial se Reconoce.</p> <p>H: Se Encuentra Frase Similar.</p> <p>I: La Frase Encontrada puede Utilizarse.</p> <p>Z: El Agente Interfaz Muestra la Señal.</p>	<p>reconocimiento de frase y asociación con frase similar.</p>	<p>reconocimiento de frase y asociación con frase similar.</p>
Caso 6	<p>A: Ingresa Frase/Palabra.</p> <p>C: La/s Palabras están Registradas.</p> <p>G: Las Palabras no se Reconocen.</p> <p>J: Se Añade Nueva Frase.</p> <p>Z: El Agente Interfaz Muestra la Señal.</p>	<p>Las palabras no se reconocen, la frase tampoco, se añade como frase nueva.</p>	<p>Las palabras no se reconocen, la frase tampoco, se añade como frase nueva.</p>
Caso 7	<p>A: Ingresa Frase/Palabra.</p> <p>B: Algunas Palabras están Registradas.</p> <p>E: Sólo la Palabra Inicial se Reconoce.</p> <p>H: Se Encuentra Frase Similar.</p> <p>I: La Frase Encontrada puede Utilizarse.</p> <p>Z: El Agente Interfaz Muestra la Señal.</p>	<p>Reconocimiento de algunas palabras, reconocimiento de frase similar y asociación con frase similar.</p>	<p>Reconocimiento de algunas palabras, reconocimiento de frase similar y asociación con frase similar.</p>
Caso 8	<p>A: Ingresa Frase/Palabra.</p> <p>B: Algunas Palabras están Registradas.</p> <p>E: Sólo la Palabra Inicial se Reconoce.</p> <p>H: Se Encuentra Frase Similar.</p>	<p>Reconocimiento de algunas palabras, reconocimiento de frase similar, pero se añade</p>	<p>Reconocimiento de algunas palabras, reconocimiento de frase similar, pero se añade</p>

	J: Se Añade Nueva Frase. Z: El Agente Interfaz Muestra la Señal.	como frase nueva.	como frase nueva.
Caso 9	A: Ingresa Frase/Palabra. B: Algunas Palabras están Registradas. G: Las Palabras no se Reconocen. J: Se Añade Nueva Frase. Z: El Agente Interfaz Muestra la Señal.	Las palabras no se reconocen, la frase no se reconoce, se añade nueva frase.	Las palabras no se reconocen, la frase no se reconoce, se añade nueva frase.
Caso 10	A: Ingresa Frase/Palabra. D: Las Palabras no están Registradas. G: Las Palabras no se Reconocen. J: Se Añade Nueva Frase. Z: El Agente Interfaz Muestra la Señal.	Las palabras no se reconocen, la frase no se reconoce, se añade nueva frase.	Las palabras no se reconocen, la frase no se reconoce, se añade nueva frase.

En las Fig. 5.2, 5.3 y 5.4 se muestran los resultados obtenidos en consola del Caso 1, 2 y 3 respectivamente.

```

Introduzca la frase:
con permiso
Palabra 1 = con
Palabra 2 = permiso

Esta es la Frase: con permiso

Palabra registrada
Palabra registrada
Palabras reconocidas
Palabras reconocidas
Existe la frase
Existe la frase
Introduzca la frase:

```

Figura 5.2 Resultados de Ejecución del Sistema del Caso 1.

<pre> Introduzca la frase: con permiso señor Palabra 1 = con Palabra 2 = permiso Palabra 3 = señor  Esta es la Frase: con permiso señor  Palabra registrada Palabra registrada Palabra registrada Palabras reconocidas Palabras reconocidas Palabras reconocidas 120: con permiso 34 136: con su ayuda 10 139: con 9 140: con su 8 142: con agua 7 152: con permiso niDo 6 163: con soda 2 165: con liquido 1 La frase con permiso señor no se encuentra DA ENTER para agregarla o si se muestra una frase similar seleccionala 34 Se añadió nueva frase </pre>	<pre> Esta es la Frase: con permiso señor  Palabra registrada Palabra registrada Palabra registrada Palabras reconocidas Palabras reconocidas Palabras reconocidas Frase similar encontrada </pre>
---	--

Figura 5.3 Resultados de Ejecución del Sistema del Caso 2.

```
Esta es la Frase: hoy nade

Palabra registrada
Palabra registrada
Palabras reconocidas
Palabras reconocidas
15: hoy tome agua 15
144: hoy tome 10
146: hoy camine 9
155: hoy tome vino 7
167: hoy corri much 3
170: hoy nade mucho 1
La frase hoy nade no se encuentra
DA ENTER para agregarla o si se
muestra una frase similar seleccionala

Se aadió nueva frase
```

*Figura 5.4 Resultados de Ejecución del Sistema del Caso 3.*

Como se observa en los últimos 3 casos, las palabras que conforman las frases se encuentran registradas y son reconocidas, en el primer caso la frase es encontrada dentro de la base de creencias, en el caso 2, la se encuentra una frase similar, a la cual se le relaciona y en el caso 3, las frases similares encontradas, no son lo suficiente similares, así que se decide por agregarla.

En la Fig. 5.5 se muestra el Caso 4, muy similar al caso 3.

```
Esta es la Frase: hoy baile

Palabra registrada
Palabra registrada
Palabras reconocidas
baile: no reconocidas en la segunda tabla, agregando
15: hoy tome agua 17
144: hoy tome 11
146: hoy camine 10
155: hoy tome vino 8
167: hoy corri much 5
170: hoy nade mucho 2
173: hoy nade 1
La frase hoy baile no se encuentra
DA ENTER para agregarla o si se
muestra una frase similar seleccionala

Se añadio nueva frase
```

*Figura 5.5 Resultados de Ejecución del Sistema del Caso 4.*

Este caso, como se mencionó anteriormente, es similar al caso 3, con la única diferencia, de no reconocer una de las palabras.

En la Fig. 5.6 se muestra el Caso 5 en la misma forma que los casos anteriores.

```

Introduzca la frase:
hoy bebi liquidos
Palabra 1 = hoy
Palabra 2 = bebi
Palabra 3 = liquidos

Esta es la Frase: hoy bebi liquidos

Palabra registrada
Palabra registrada
Una o más palabras no registrada
Palabra registrada
Palabras reconocidas
Palabras reconocidas
liquidos: no reconocidas en la segunda tabla, agregando
15: hoy tome agua 6
144: hoy tome 2
146: hoy camine 1
La frase hoy bebi liquidos no se encuentra
DA ENTER para agregarla o si se
muestra una frase similar seleccionala
15
Se añadio nueva frase

Esta es la Frase: hoy bebi liquidos

Palabra registrada
Palabra registrada
Palabra registrada
Palabras reconocidas
Palabras reconocidas
Palabras reconocidas
Frase similar encontrada

```

Figura 5.6 Resultados de Ejecución del Sistema del Caso 5.

Este caso, las palabras que conforman la frase se encuentran registradas, pero solo la palabra inicial es reconocida, por lo tanto, la frase no se encuentra en la base de creencias, y de manera similar al caso 2, hay que asociar la frase a una ya existente en caso de exista una.

En la Fig. 5.7 se muestran los resultados del Caso 6.

```

Esta es la Frase: senora camine

Palabra registrada
Palabra registrada
senora: no reconocidas en la segunda tabla, agregando
camine: no reconocidas en la segunda tabla, agregando
La frase senora camine no se encuentra
DA ENTER para agregarla o si se
muestra una frase similar seleccionala

Se añadio nueva frase

```

Figura 5.7 Resultados de Ejecución del Sistema del Caso 6.

Como se observa, este caso es algo sencillo, las palabras se encuentran registradas, más no son reconocidas, por lo tanto, no existe la frase y no existe frase similar, por lo que solo queda añadirla.

En la Fig. 5.8 se muestran los resultados para el Caso 7.

```
Introduzca la frase:                Esta es la Frase: con permiso doctor
con permiso doctor
Palabra 1 = con                      Palabra registrada
Palabra 2 = permiso                 Palabra registrada
Palabra 3 = doctor                  Palabra registrada
Esta es la Frase: con permiso doctor Palabras reconocidas
Palabra registrada                  Palabras reconocidas
Palabra registrada                  Palabras reconocidas
Una o más palabras no registrada    Frase similar encontrada
Palabra registrada
Palabras reconocidas
Palabras reconocidas
doctor: no reconocidas en la segunda tabla, agregando.
120: con permiso 36
136: con su ayuda 12
139: con 11
140: con su 10
142: con agua 9
152: con permiso niDo 8
163: con soda 4
165: con liquido 3
La frase con permiso doctor no se encuentra
DA ENTER para agregarla o si se
muestra una frase similar seleccionala
120
Se añadió nueva frase
```

Figura 5.8 Resultados de Ejecución del Sistema del Caso 7.

Al igual que en casos anteriores, no se cumplieron las condiciones para poder asociar la frase con alguna similar, por lo que se decide agregarla como una nueva.

Las figuras 5.9, 5.10 y 5.11, corresponden a los casos 8,9 y 10 respectivamente.

```
Introduzca la frase:
hoy tome vino
Palabra 1 = hoy
Palabra 2 = tome
Palabra 3 = vino

Esta es la Frase: hoy tome vino

Palabra registrada
Palabra registrada
Una o más palabras no registrada
Palabra registrada
Palabras reconocidas
Palabras reconocidas
vino: no reconocidas en la segunda tabla, agregando.
15: hoy tome agua 8
144: hoy tome 3
146: hoy camine 2
La frase hoy tome vino no se encuentra
DA ENTER para agregarla o si se
muestra una frase similar seleccionala

Se añadió nueva frase
```

Figura 5.9 Resultados de Ejecución del Sistema del Caso 8.

```
Introduzca la frase:
ire por vino
Palabra 1 = ire
Palabra 2 = por
Palabra 3 = vino

Esta es la Frase: ire por vino

Una o más palabras no registrada
Palabra registrada
Palabra registrada
Palabra registrada
ire: no reconocidas en la segunda tabla, agregando..
por: no reconocidas en la segunda tabla, agregando..
vino: no reconocidas en la segunda tabla, agregando.
La frase ire por vino no se encuentra
DA ENTER para agregarla o si se
muestra una frase similar seleccionala

Se añadió nueva frase
```

Figura 5.10 Resultados de Ejecución del Sistema del Caso 9.

```
Introduzca la frase:
libro dificil
Palabra 1 = libro
Palabra 2 = dificil

Esta es la Frase: libro dificil

Una o más palabras no registrada
Palabra registrada
Una o más palabras no registrada
Palabra registrada
libro: no reconocidas en la segunda tabla, agregando..
dificil: no reconocidas en la segunda tabla, agregando
La frase libro dificil no se encuentra
DA ENTER para agregarla o si se
muestra una frase similar seleccionala

Se añadio nueva frase
```

*Figura 5.11 Resultados de Ejecución del Sistema del Caso 10.*

Como se muestra en la Fig. 5.9, no todas las palabras están registradas y no todas fueron reconocidas y aunque hubo frases similares, se decidió por añadirla como nueva, esta misma situación se muestra en las siguientes dos figuras, pues en ambas las palabras no son reconocidas y en una de ellas ni si quiera se están registradas, por lo que la frase resultante no se encontrará en la base de creencias y no es posible encontrar alguna frase similar, por lo cual la frase debe ser añadida.

Como se observó en los 10 casos anteriores, hay 3 posibilidades, existe la frase, no existe o existe una similar, para estas dos últimas posibilidades es necesaria la intervención del usuario, aunque dicha acción podría ser realizada por los agentes encargados del sistema, sugerencia que se mencionará en el siguiente capítulo.

Dependiendo de la última acción tomada, en estos casos, será la acción que realizará el agente Interfaz, siendo que al existir la frase, también existe una seña asociada a ella, al existir una frase similar, se utiliza la seña original de dicha frase similar o exactamente de la ingresada, pero resultar todo en una nueva frase, la seña resultante dependerá de la nueva frase ingresada, pudiendo ser que si alguna de las palabras se encuentran registradas y/o son reconocidas, exista una seña que las defina a ambas en conjunto o deban ser definidas individualmente y la palabra no registrada tenga que ser deletreada. En caso que solo una de las palabras se encuentre, esta será definida individualmente y las demás tendrán que ser deletreadas y en caso de ser todas palabras nuevas, deberán ser deletreadas todas.

En caso que se agregue una nueva frase con palabras ya registradas o incluso reconocidas, se podrán concatenar las señas que definen a cada una de estas palabras o a su conjunto.

Continuando con los resultados, el IDE de JADE contiene una herramienta llamada sniffer, la cual nos permite visualizar la interacción entre agentes, en la Fig. 5.12, se muestra como fue dicha interacción entre el agente Coordinador y el agente Interfaz.

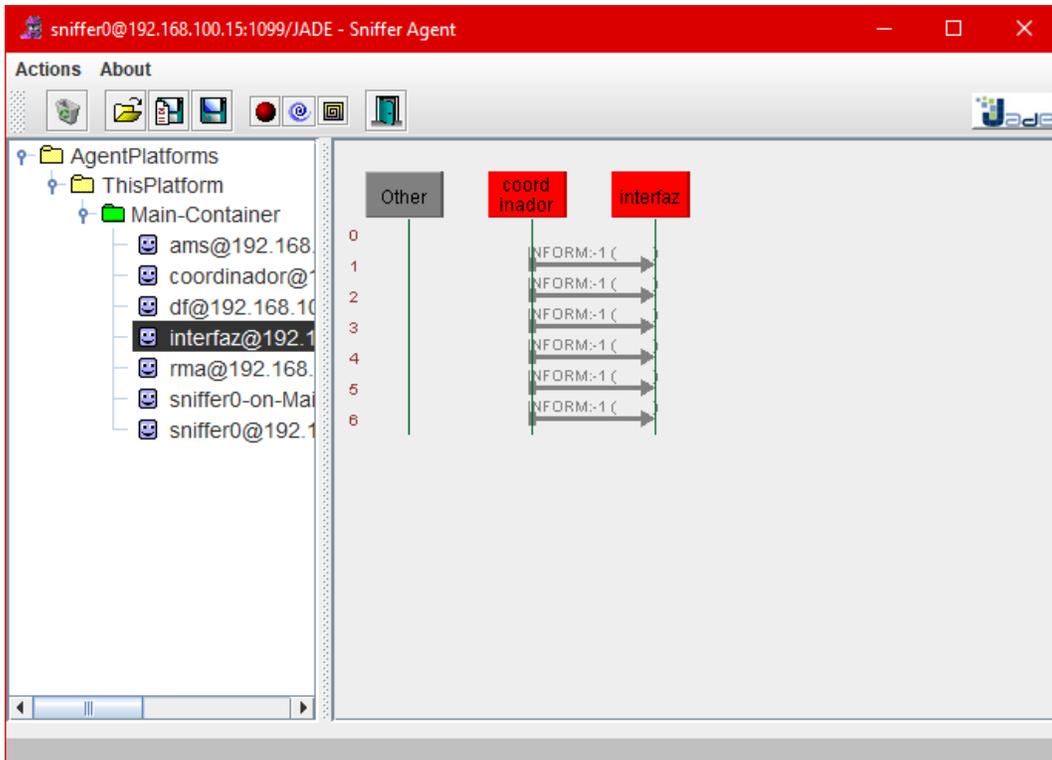


Figura 5.12 Interacción Entre Agentes

La interacción mostrada en la figura 5.12, es muy básica, siendo esta en un solo sentido, puesto que solo el agente Coordinador envía y el agente Interfaz recibe, ocurriendo así el mismo tipo de interacción en todos los casos, variando solamente en el mensaje enviado. En la Fig. 5.13, se muestra la información que fue enviada, cual agente es el que envió el mensaje y cual agente lo recibió.

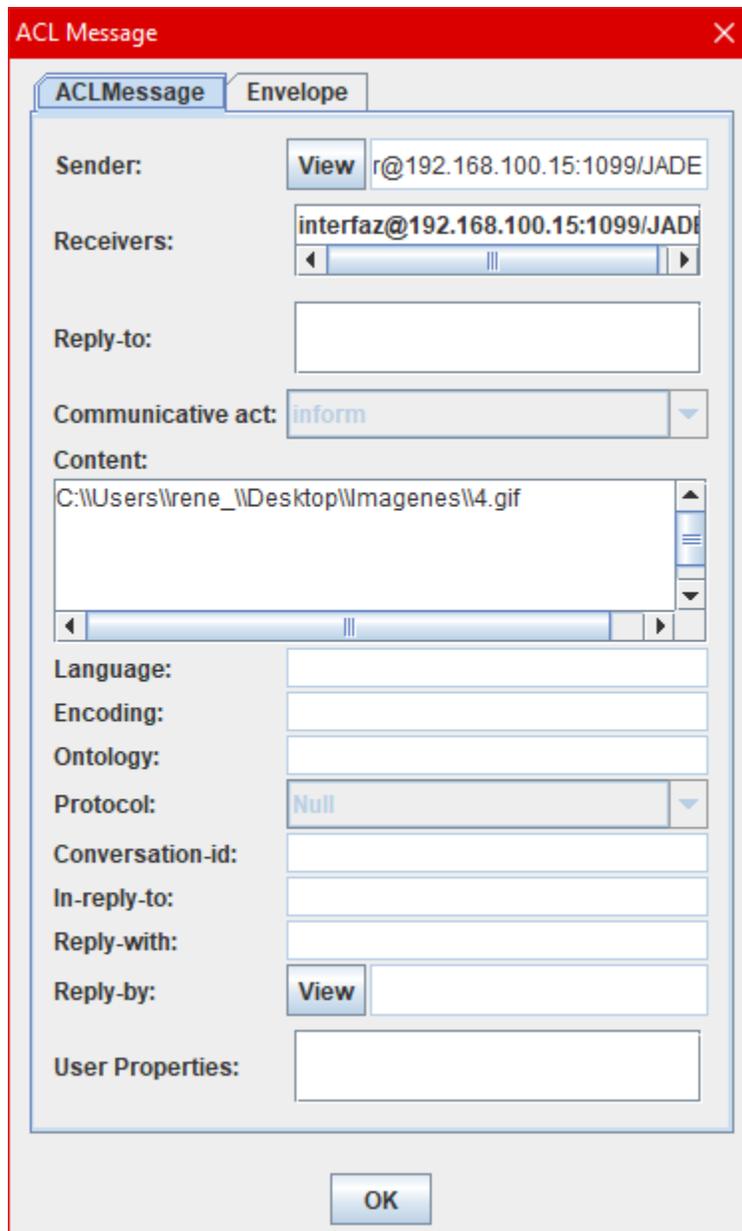


Figura 5.13 Información del Mensaje Entre Agentes

Como se observa en la figura anterior, en la casilla "Content", se muestra un mensaje, el cual corresponde a una dirección, misma que contiene la animación o imagen a representar por el agente una vez recibe el mensaje del agente Coordinador, la cual se observa en la Fig. 5.14.

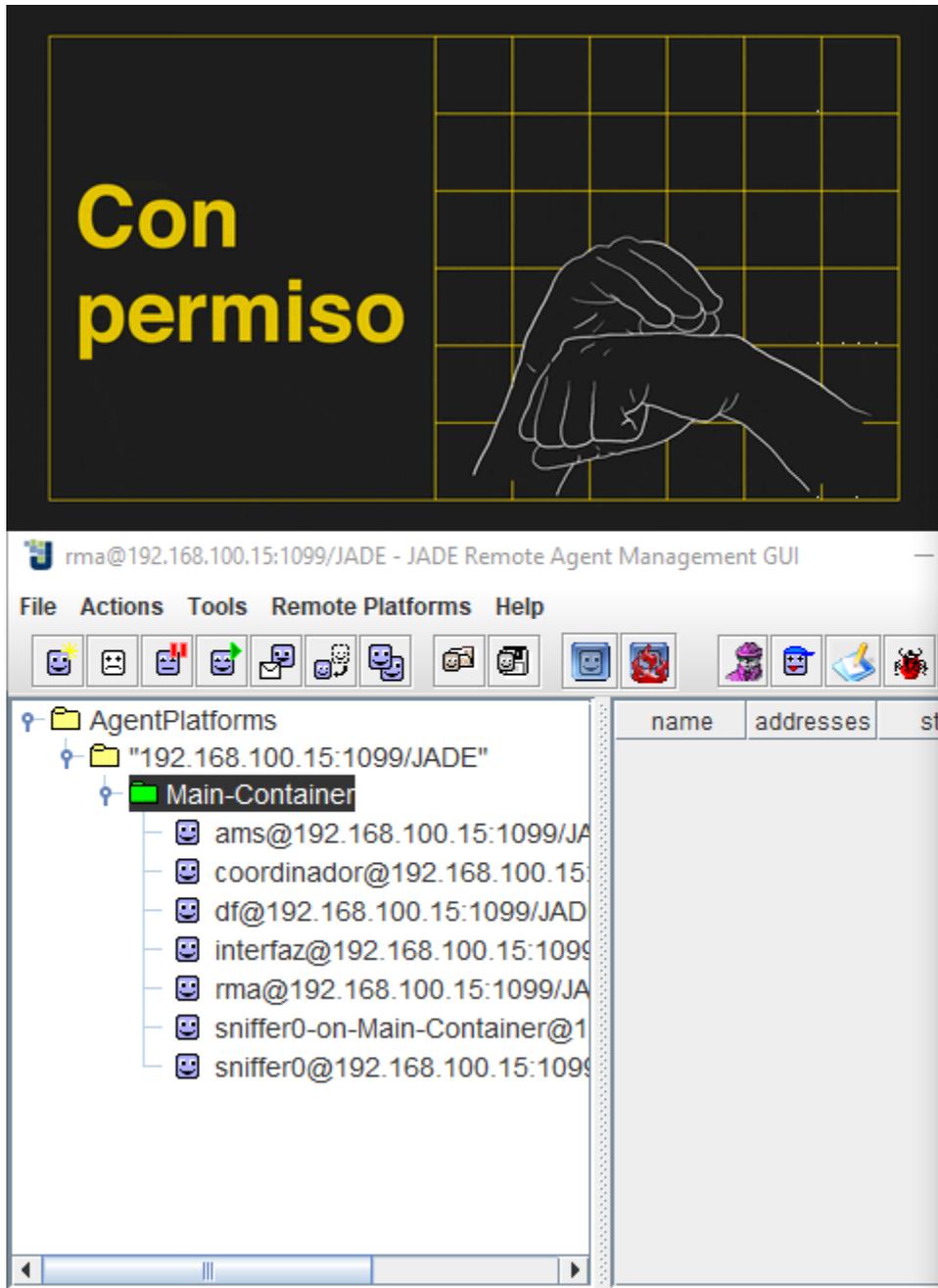


Figura 5.14 Resultado Final al Recibir Frase

## Capítulo 6: Conclusiones y Trabajo Futuro

En el presente capítulo se hablará de muchas de las decisiones tomadas al realizar y desarrollar este proyecto, pues aunque se obtuvieron resultados, el enfoque que inicialmente se le había dado, era un poco distinto, por lo cual también se hablará

de una posible continuación del proyecto o combinación de varios de estos, en donde se mencionarán los posibles alcances del mismo, que nuevas características podrían ser agregadas y como podría resultar en un sistema más completo y exacto, donde se tomaran en cuenta muchos más escenarios de los que se tomaron originalmente.

## **6.1. Conclusiones**

En este trabajo se logró desarrollar un modelo, capaz de reconocer entre varias frases la traducción que más se adapte a lo buscado, por último la pregunta, utilizando de manera fija una de las formas gramaticales del LSM, logrando que todas las frases sean uniformes en cuanto a su estructura.

Los objetivos presentados al inicio de este proyecto de igual manera se cumplieron, logrando diseñar un modelo capaz de reconocer y representar frases en LSM utilizando CBR en conjunto con agentes inteligentes.

Trabajar con frases de un idioma como el español no es una tarea sencilla, existen muchas variaciones en el mismo, demasiadas reglas gramaticales, aparte de no ser un idioma 100% estructurado, pues dependiendo de la situación, el contexto, la región, incluso dentro de un mismo país, una misma frase o palabra, pueden significar muchas cosas completamente diferentes, sucediendo también a la inversa, siendo que muchas frases o palabras diferentes entre sí, pueden tener el mismo significado, por lo tanto realizar una traducción o interpretación precisa, muchas veces se vuelve una tarea imposible, pues son muchos los factores a tomar en cuenta al momento de querer interpretar alguna oración. Trabajar solo con este idioma ya es complicado, ahora traducirlo a un lenguaje de señas, derivado de otro idioma, en este caso del francés, vuelve a la tarea aún más complicada, pues el lenguaje de señas mexicano, está adaptado a su idioma original y no al español, como era de esperarse, complicando así un poco más el proceso de traducción, aparte de las propias reglas del mismo lenguaje que

dependiendo de la situación en la que se encuentre una sola seña puede significar una frase completa o conjunto de estas alguna acción en concreto.

En este proyecto se decidió por abarcar de manera general frases sencillas del español, aparte de considerarlas acomodadas en gramática de señas mexicana, utilizando ejemplos de un máximo de 4 palabras (aunque pudieran utilizarse más), y aunque se obtuvieron resultados, aun el proyecto se encuentra lejos de ser un traductor a lenguaje de señas completo, se necesitaría un análisis más a fondo del español y quizá la inclusión de técnicas más complejas de inteligencia artificial, aparte de la adición de más comportamientos y creencias a los agentes y en conjunto con el sistema CBR se pudieran obtener mucho mejores resultados, de manera más autónoma de cómo se han logrado hasta el momento.

Ampliar más aun este proyecto y lograr que funcione completamente por sí mismo, ayudaría a tener una aplicación muy útil, para las personas a quienes va dirigido este trabajo, los Sordos, consiguiendo así mejorar su calidad de vida, que como se vio en el capítulo 1, esta es muy complicada y deficiente, por el poco apoyo que reciben.

## **6.2. Trabajo Futuro**

Como se ha mencionado en las conclusiones, una propuesta de continuación al proyecto sería implementar más comportamientos a los agentes o incluso añadir un tercer agente, y entre ellos poder interpretar inicialmente las frases y comprender en qué contexto se están utilizando, para posteriormente procesarlas y obtener un resultado más acertado. También como se mostró en los ejemplos de interpretación de resultados, la intervención del usuario era necesaria, para poder seleccionar que sucedía con una frase que encontraba otras similares o cuando no encontraba nada, así como también la búsqueda o creación y relación de las señas necesarias correspondientes a las palabras y/o frases, estas acciones pudieran ser realizadas por los mismos agentes, siempre y cuando estos contaran con los comportamientos necesarios para tomar las decisiones correctas, incluso

existiendo la posibilidad de consultar bases de datos externas al sistema y poder utilizarlas dentro del mismo, ampliando así la utilidad de esta aplicación.

Otro plan de trabajo y uno de los motivos por los cuales se decidió realizar así el presente trabajo, fue la consideración del trabajo del M.C.C. O. Carvajal, "Sistema de reconocimiento de voz y traducción de lengua de señas mediante un avatar", el cual consiste, en recibir una frase o palabra y por medio de reconocimiento de voz se ingresa a su sistema, se migra de la gramática original de la frase a gramática de lenguaje de señas mexicano y mediante un algoritmo diseñado en dicho trabajo, sea crea una animación en 3D, la cual interpreta la frase en lenguaje de señas mexicano y se muestra en la interfaz de usuario. Partiendo de este trabajo y en conjunto con el proyecto actual, se lograría obtener un sistema mucho más eficiente, autónomo y completo, gracias a la complementación de un sistema con el otro, puesto que no se tendrían que crear las animaciones cada vez se ingresara una frase ya utilizada, ni se tendrían que crear animaciones innecesarias en situaciones donde una frase ingresada pudiera funcionar con otra similar, todo esto y aplicando lo mencionado anteriormente, sobre mejorar los comportamientos de los agentes, convertiría el proyecto en una aplicación sumamente útil, para las personas a quienes va dirigida.

## Referencias

- [1] Teresa C, Tania C, Jesús V, Evaluación del paciente con Hipoacusia. Madrid, España, Capítulo 32.
- [2] “Comunidad sordomuda” <http://www.who.int/mediacentre/factsheets/fs300/es/> (Visitado el 27 de octubre del 2017).
- [3] “Comunidad sordomuda en México” <https://sipse.com/mexico/sordos-discapacidad-gobierno-mexico-224324.html> (Visitado el 27 de octubre del 2017).
- [4] “Discapacidad en México” <http://cuentame.inegi.org.mx/poblacion/discapacidad.aspx?tema=P> (Visitado 19 de noviembre de 2017).
- [5] NICHCY (2010). La Sordera y la Pérdida de la Capacidad Auditiva. Estados Unidos. [En línea]. Disponible: <https://www.isbe.net/Documents/deafness-hearing-loss-sp.pdf> (Visitado 09 de febrero del 2020).
- [6] “Sordera y pérdida de la audición” [En línea]. Disponible: <https://www.who.int/es/news-room/fact-sheets/detail/deafness-and-hearing-loss> (Visitado 09 de febrero del 2020).
- [7] “Lenguaje de señas y sus distintas formas de comunicar” <https://www.infobae.com/america/mexico/2019/09/25/el-lenguaje-de-senas-y-sus-distintas-formas-de-comunicar/> (Visitado 09 de febrero del 2020).
- [8] Pablo A. Espinosa A. & Hernán A. Pogo L. (2013). Diseño y construcción de un guante prototipo electrónico capaz de traducir el lenguaje de señas de una persona sordomuda al lenguaje de letras. [En línea]. Disponible: <https://dspace.ups.edu.ec/bitstream/123456789/4211/1/UPS-CT002598.pdf>.
- [9] Carvajal, O. E. (2018). Sistema de reconocimiento de voz y traducción de lengua de señas mediante un avatar. México, Hermosillo.
- [10] Maria Esther Serafin de Fleischmann, R. G. (2011). Manos con voz. México: Libre Acceso A.C.

- [11] Kabil Jaballah, M. J. (2012). Sign Language parameters classification from 3D. International Conference on Information Technology and e-Services, (pág. 6). Tunisia.
- [12] Aldrete, M. C. (2008). Gramática de la lengua de señas mexicana. México.
- [13] Paolo Priore Moreno, J. P. (2014). Utilización del Razonamiento Basado en Casos en la toma de decisiones. Aplicación en un problema de secuenciación. 14.
- [14] Dapena, M. D. (2003). Calidad de los proyectos de software: Revisiones utilizando Razonamiento Basado en Casos. 8.
- [15] Gómez, F. D. (s.f.). Razonamiento Basado en Casos. Valladolid, Segovia, España.
- [16] Augusto Cortez Vásquez, C. N. (2010). Sistemas de Razonamiento Basado en casos aplicado a sistemas de líneas de productos software. Revista de Investigación de Sistemas e Informática, 6.
- [17] Wooldridge, M. (2009). MultiAgent Systems. Torquay: Wiley.
- [18] Norving, S. R. (2010). Artificial Intelligence A Modern Approach. Upper Saddle River: Pearson Education.
- [19] Weiss, G. (1999). Multiagent Systems A Modern Approach to Distributed Modern Approach to Artificial Intelligence. Londres.
- [20] Mas, A. (2004). Agentes software y sistemas multiagente: conceptos, arquitecturas y aplicaciones. Pearson, Madrid, España.
- [21] Wooldrige M, Jennings, Nicholas. (1994). Lecture Notes in Artificial Intelligence. Springer-Verlag. Amsterdam, Países Bajos.
- [22] M.P Carretero, D. O. (2004). Animación Facial y Corporal de Avatares 3D a partir de la Edición e Interpretación de Lenguaje de Marcas. 12.
- [23] Jing Wang, Y. S. (2010). Chinese Sign Language Animation System on Mobile Devices. 4.

[24] Dailey, L. (February de 2008). IBM System is a Virtual Sign-Language Interpreter. News Briefs, pág. 3.

[25] Bautista, V., Robayo, E. (2019). Modelo ISO/IEC 25010 en el proceso de evaluación de la calidad del software en la empresa Obras Civiles de Bogotá en el área de tecnología de la información y comunicación. Bogotá.

[26] Sánchez. J. (2015). Pruebas de Software. Fundamentos y Técnicas. Madrid.