



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

SISTEMA EXPERTO PARA RECONOCIMIENTO DE
LENGUAJE NATURAL CON ESPECIALIZACIÓN EN TÉRMINOS
DE LA CONSTITUCIÓN

Carlos Estuardo Coy de León

Asesorado por Inga. Marcela Velásquez

Guatemala, noviembre de 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMA EXPERTO PARA RECONOCIMIENTO DE
LENGUAJE NATURAL CON ESPECIALIZACIÓN EN TÉRMINOS
DE LA CONSTITUCIÓN**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR
CARLOS ESTUARDO COY DE LEÓN

ASESORADO POR INGA. MARCELA VELAZQUEZ

AL CONFERIRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXÁMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Inga. Vivian Damaris del Socorro Campos Gonzalez
EXAMINADOR	Ing. Edgar René Ornelis Hoil
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

SISTEMA EXPERTO PARA RECONOCIMIENTO DE LENGUAJE NATURAL CON ESPECIALIZACIÓN EN TÉRMINOS DE LA CONSTITUCIÓN

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 19 de julio de 2004.

Carlos Estuardo Coy de León

Agradecimientos

A **Dios** quien es la fuente de mi fortaleza y es mi todo, por haberme dado fuerzas en cada paso que doy, por ser mi principio y no dejarme sin fin.

A **mi papá** por enseñarme por medio de su ejemplo innegable y constancia en su caminar, que se puede salir adelante a pesar de la adversidad y de las circunstancias que nos rodean, y por haber soñado el camino a seguir para lograr mis metas y apoyarme en cada paso que doy hasta alcanzar nuestro sueño.

A **mi mamá** que desde niño me hizo comprender que nada hay imposible para el que puede creer y que todo lo puedo en Cristo que me fortalece, por haberme enseñado el camino, la verdad y la vida y haberme guiado hacia ese camino, por esas horas de interminable dedicación a su hijo y por haber formado un hombre que no se puede rendir ante nada.

A **mis hermana Karen** que desde que nació ha sido un regalo del cielo, por mostrarme con su valentía y dedicación en todo lo que emprende, que se puede enderezar el curso de nuestra vida aún cuando nos desviamos.

A **mi hermano Gerson** que a pesar el menor de todos es más fuerte que yo. Por apoyarme con su gran corazón y cariño, por desvelarse junto conmigo y ser mi amiguito. Por querer aprender lo bueno, apoyar las causas nobles y tomar visión para lograr sus sueños, eso es inspirador.

A **Mónica** que ha sabido llegar al fondo de mi corazón y sacar lo mejor del tesoro de mi corazón y pasar a ser parte de ese tesoro, por querer ser el instrumento que me inspira a no desmayar, a buscar todo lo que es bueno, justo y agradable, y por haberme enseñado a sonreír aún cuando las circunstancias no lo ameritan.

A **mis amigos** Julio, Carlos, Roberto, Otto, Danilo, Ronald, Henry, Rene, Norma por haberse desvelado luchando y por saber alegrarse junto conmigo en el esfuerzo final.

Al movimiento **hálito** y los amigos que allí he hallado que son de incalculable estima más que el oro.

A mi asesora la **Inga. Marcela Velásquez** quien con su diligencia y firme apoyo me a ayudado a concluir este trabajo de graduación con satisfacción.

Al **Ministerio de Educación** por haberme abierto sus puertas y especialmente al personal de la UDI y DICADE que han seguido de cerca cada paso final que he dado para alcanzar mi sueño.

A las iglesias **Principe de Paz central** y **Jesucristo es el Señor zona 21**, por ayudarme a formar mi vida espiritual y enseñarme a hacer prosperar mi alma, para así, prosperar en todo lo que hago por medio de la única verdad que es mi salvador Jesucristo.

Y a **Dios** doy infinitas gracias, porque el es mi fin, porque sus planes para mi vida son planes de bien y no de mal para darme el fin que espero y porque las promesas y llamamiento de Dios son irrevocables y me ha dado muestras de ello, por haberme dado un Salvador en Jesucristo que ha sido mi escondedero fuerte en quien puedo confiar sin dudas y por haber puesto a todas esas personas importantes en mi vida, porque han sido necesarias para darme aliento y amor. Gracias Abba Padre.

ÍNDICE

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	VII
RESUMEN	XIII
OBJETIVOS	XV
INTRODUCCIÓN	XVII
1 MARCO TEÓRICO	1
1.1 Introducción.....	1
1.2 Antecedentes.....	1
1.2.1 Test de Turing.....	2
1.2.2 A.L.I.C.E	2
1.3 Reconocimiento de Lenguaje Natural.....	3
1.3.1 Lenguaje Natural	3
1.3.2 Lenguajes formales	4
1.3.2.1 Características de los lenguajes Formales.....	4
1.3.3 Aplicaciones del Reconocimiento de Lenguaje Natural.....	5
1.4 Traducción Automática.....	6
1.4.1 Metodologías básicas utilizadas	7
1.4.2 El análisis sintáctico.....	9
1.4.3 Sistemas basados en Conocimiento.....	10
1.5 Conclusiones	11
2 ENTRADAS DEL SISTEMA	13
2.1 Análisis	13
2.1.1 Requerimientos Funcionales	13
2.1.2 Límites	14
2.1.2.1 Conocimiento	14
2.1.2.2 Estructura.....	15

2.1.3	Alcances.....	16
2.1.4	Lengua Origen (LO)	17
2.1.5	Conceptos del Lenguaje Natural	18
2.1.5.1	Clasificación de las palabras	19
2.2	Proceso de Reconocimiento	21
2.2.1	Análisis Léxico.....	21
2.2.2	Análisis Sintáctico	23
2.2.2.1	Emparejamiento de Patrones (Eliza)	25
3	SALIDAS DEL SISTEMA	27
3.1	Interpretación de Oraciones	29
4	CONSTRUCCIÓN DEL SISTEMA	33
4.1	Lexicón.....	33
4.1.1	Palabra.....	34
4.1.2	Artículo.....	34
4.1.3	Sustantivo.....	35
4.1.4	Preposición.....	35
4.1.5	Determinante.....	36
4.1.6	Verbo.....	36
4.1.7	VERBO_REGULAR	37
4.1.8	VERBO_IRREGULAR.....	37
4.1.9	Conjunción	38
4.1.10	Adverbio	38
4.2	Hechos y variables globales.....	40
4.2.1	Hechos	40
4.2.1.1	Definición de Hechos.....	40
4.2.1.2	Definición de Plantillas.....	41
4.3	Reglas.....	44
4.3.1	Inicio.....	44
4.3.2	reconoce_articulo.....	45
4.3.3	reconoce_preposicion	45

4.3.4	reconoce_verbo_irregular_presente	46
4.3.5	reconoce_verbo_irregular_pasado	47
4.3.6	reconoce_verbo_regular	47
4.3.7	reconoce_sustantivo.....	48
4.3.8	reconoce_adverbio	49
4.4	Proceso de equiparación de reglas	50
4.5	Aprendizaje del SRLN	53
4.5.1	Módulo de ingreso y consulta de nuevas palabras	54
4.5.1.1	Ingreso de nuevas palabras en el lexicón	55
4.5.1.2	Consulta de palabras del lexicón	56
4.6	Ejemplos de Ejecución	58
4.6.1	Ejemplo de Oración de Entrada Afirmativa.....	58
4.6.2	Ejemplo de Oración de Entrada con Negación.....	60
5	FUNDAMENTOS DE USO, FUNCIONAMIENTO Y APORTES	63
5.1	Requerimientos	63
5.1.1	Requerimientos de Hardware	64
5.1.2	Requerimientos de Software	64
5.1.3	Recursos Humanos	65
5.2	Límites y Alcances.....	65
5.2.1	Límites	66
5.2.2	Alcances	66
5.3	Impacto esperado y Aportes del SRLN.....	67
	CONCLUSIONES.....	69
	RECOMENDACIONES	71
	BIBLIOGRAFÍA ELECTRÓNICA	73
	APÉNDICE	75

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Proceso de comunicación	18
2. Proceso del Análisis Léxico	22
3. Análisis Sintáctico	27
4. Transformación de mensajes	30
5. Diagrama de clases de palabras	39
6. Diagrama de estados del proceso de equiparación de reglas	50
7. Pantalla de ingreso de Propuestas	52
8. Ingreso y Consulta de nuevas palabras	54
9. Ingreso de un artículo al lexicón	56
10. Pantalla de resultado para los verbos irregulares	57
11. Consulta de un artículo ingresado por el usuario	58
12. Pantalla de ingreso de propuestas (Ejemplo 1)	59
13. Aviso de finalización de análisis	59
14. Pantalla de ingreso de propuestas (Ejemplo 2)	61

TABLAS

I. Tipos de Adverbios	20
II. Tipos de Artículos	20
III. Tipos de determinantes	20

GLOSARIO

- Ambigüedad** La ambigüedad semántica se refiere a los muchos significados que puede tomar una palabra y que, por consiguiente, puede alterar el significado de una oración. De ser posible, se elige el significado de acuerdo al contexto en el cual es utilizada la palabra.
- Clase** Serie de normas o especificaciones que definen el comportamiento de un objeto dado, indicando las propiedad y funciones que tendrá cada objeto. Una Clase es utilizada en la programación orientada a objetos. Un Diagrama de Clases, entonces, define un diseño de la jerarquía y relaciones que existen entre diferentes clases que formarán parte de un sistema de cómputo.
- Compilador** Un compilador acepta programas escritos en un lenguaje de alto nivel y los traduce a otro lenguaje, generando un programa equivalente, independiente, que puede ejecutarse tantas veces como se quiera. Este proceso de traducción se conoce como compilación.
- EHSIS** Generador de Sistemas Expertos que cuenta con un lenguaje de programación para tal efecto.

Fonema	Término usado en lingüística para referirse a cualquier sonido del habla que permite distinguir palabras en una lengua.
Fonología	Rama de la lingüística que estudia los fonemas.
Gramática	Estudio de la lengua, en cuanto a forma, estructura, y significado.
Instancia	Una instancia es una llamada a una clase, lo cual forma un nuevo objeto que contiene las propiedades y funcionalidad expresadas en una clase.
Léxico	Conjunto de palabras de un idioma, de una región de este, o incluso de un lenguaje de programación.
Morfología	Estudio de las distintas formas o variantes del significante de las palabras de una lengua
Ontología	<p>El término ontología, en informática hace referencia al intento de formular un exhaustivo y riguroso esquema conceptual dentro de un dominio dado con la finalidad de facilitar la comunicación y compartir la información entre diferentes sistemas.</p> <p>Típicamente, las ontologías en los ordenadores se relacionan estrechamente con vocabularios fijos, con cuyos términos debe ser descrito todo lo demás.</p>

Polisemántica	Posibilidad de que una misma oración tenga más de un significado.
Procesamiento de lenguajes naturales	(NLP, Natural Language Processing en ingles). Es una subdisciplina de la Inteligencia Artificial y, también de la lingüística computacional. Estudia los problemas inherentes al procesamiento y manipulación de lenguajes naturales, sin embargo no suele plantear el entendimiento de lenguajes naturales.
Prototipo	Cualquier tipo de máquina en pruebas, o un programa diseñado para una demostración de cualquier tipo. Este tipo de prototipos permiten probar el objeto antes de que entre en producción, detectar errores, deficiencias, etc. Cuando el prototipo está suficientemente perfeccionado en todos los sentidos requeridos y alcanza las metas para las que fue pensado, el objeto puede empezar a producirse.
Semántica	Es un subcampo de la gramática y, por extensión, de la lingüística. Proviene del griego <i>semantikos</i> que quiere decir significado relevante, derivada de <i>sema</i> , lo que significa signo. Se dedica al estudio del significado de los signos lingüísticos y de sus combinaciones.
Sintagma	Estructura sintáctica en la que no existe la relación de sujeto y el predicado y consiste en un conjunto de palabras relacionadas con un núcleo que se encuentra en el interior de tal sintagma.

El sintagma posee además una función sintáctica en su contexto y, a diferencia de la oración, no posee una entonación específica, al menos en español. Todas las oraciones están compuestas por sintagmas.

Sintaxis

Parte de la gramática que se encarga de estudiar las reglas que gobiernan la forma en que las palabras se organizan en sintagmas y, a su vez, estos sintagmas en oraciones.

Sistema Experto

Son sistemas expertos aquellos programas que se realizan haciendo explícito el conocimiento en ellos, que tienen información específica de un dominio concreto y que realizan una tarea relativa a este dominio. El conocimiento del que hacen uso los sistemas expertos, es conocimiento relativo a expertos humanos los cuales alimentan de información y conocimiento a dichos sistemas, es ese el punto culminante de un Sistema Experto, simular el conocimiento y experiencia de un humano en una materia en particular.

Traducción automática

Es aquella traducción que se hace por medio de ordenadores. Este sistema, en general, falla bastante y no se obtiene traducciones de calidad comparable a la que puede dar una traducción realizada por un ser humano. No obstante, son un importante avance en la comunicación, ya que, permiten la comprensión de artículos escritos por personas que no comparten los mismos idiomas hablados.

Turing

Alan Mathison Turing (23 de junio de 1912 - 7 de junio de 1954). Fue matemático, científico de la computación, criptógrafo y filósofo. Se le considera uno de los padres de la Ingeniería informática siendo el precursor de la Ciencia de la computación moderna. Proporcionó una influyente formalización de los conceptos de algoritmo y computación: la máquina de Turing.

ViaVoice

Sistema de cómputo que permite al usuario interactuar con la computadora por medio del habla humana (voz).

RESUMEN

De la inteligencia artificial se espera el desarrollo de sistemas de cómputo o máquinas que emulen el comportamiento humano. Parte esencial del comportamiento humano, es el lenguaje natural, el cual, por supuesto, ha sido emulado en múltiples sistemas de cómputo por medio de los lenguajes formales. Una de las ramas del reconocimiento de lenguaje natural lo constituye la traducción automática, la cual ha sido, ampliamente, utilizada para traducción de textos de un idioma a otro, principalmente, ésta, a su vez, hace uso del análisis léxico, sintáctico y del conocimiento del lenguaje a través de bases de hechos, más conocidas para tal efecto como lexicón. Es esta rama del reconocimiento del lenguaje natural, la que se ha utilizado como el elemento base para la construcción de un prototipo de Sistema Experto para Reconocimiento de Lenguaje Natural (SRLN) en este trabajo de graduación.

Para la construcción de todo sistema, es necesario definir las entradas y las salidas del mismo. En este trabajo las entradas del sistema las constituyen las frases del usuario experto, es decir, la lengua origen (LO) las cuales están formadas por las palabras propias del lenguaje natural. Las palabras están clasificadas por tipos de palabras, según la función que desempeñan en una oración, a fin de realizar un análisis de las palabras válidas - análisis léxico - y ordenamiento - análisis sintáctico - correcto de las mismas.

Las salidas de un sistema son la parte más importante para los usuarios, ya que es de esta manera que obtienen los resultados que esperan a fin de retroalimentarse o desempeñar una tarea. En este trabajo, las salidas retroalimentan a otro sistema experto, el cual debe ser capaz de leer una gramática definida dentro del SRLN, se espera de cualquier sistema – usuario - que interactúe con el sistema descrito en este trabajo de graduación, que pueda reconocer una estructura de oración que se define por medio de objetos.

La construcción del SRLN se basó en un elemento básico, el cual es el lexicón. El lexicón se ayuda de la programación orientada a objetos en EHSIS, a fin de almacenar en la base de hechos, las palabras propias del lenguaje natural y sus características - según la clasificación de cada palabra - y es en base al lexicón sobre el que se realiza el análisis léxico y parte del análisis sintáctico en el SRLN.

En este trabajo de graduación se pone especial énfasis en términos de la constitución de la República de Guatemala, de tal manera que el SRLN, sea capaz de reconocer un lenguaje natural basado en la constitución y que, además, de reconocerlo, pueda traducirlo a un conjunto de salidas, que sean utilizadas por otro sistema experto que se especialice en el análisis de propuestas de ley.

OBJETIVOS

General

Desarrollar un prototipo de Sistema Experto de Reconocimiento de Lenguaje Natural que traduzca la fraseología propia de un Experto Legislativo humano - idioma español - hacia un conjunto de patrones que sean, fácilmente, reconocidos por un Sistema Experto Legislativo – marcos -.

Específicos

1. Crear un Diccionario, Base de Hechos, que contenga componentes léxicos comúnmente utilizados en la Constitución de la República de Guatemala, artículos 1 al 10, y componentes comunes del idioma español.
2. Definir un lenguaje meta (LM) que sirva como entradas para la alimentación de un Sistema Experto Legislativo.
3. Diseñar, construir, y probar el Sistema orientado hacia el reconocimiento de propuestas de ley.
4. Reconocer componentes léxicos utilizando un Diccionario de términos y a través de análisis sintáctico de frases.

INTRODUCCIÓN

El lenguaje natural es un elemento muy común entre las personas, de tal manera que es utilizado, constantemente, para comunicarse entre sí, sin importar el idioma, el lenguaje natural es fácilmente expresado y comprendido por cualquier persona siendo este el medio principal por medio del cual se expresa todo ser humano.

El procesamiento del lenguaje natural por medios computacionales, durante mucho tiempo ha sido un área en la cual muchos expertos han disertado sin llegar a realizar premisas sólidas del método que se debe utilizar para el reconocimiento del lenguaje natural.

El reconocimiento de lenguaje natural que ha de realizarse con el prototipo de este trabajo de graduación, consiste en la traducción de un texto expresado en términos del idioma español, que describa una propuesta de ley, para que, posteriormente, pueda proporcionar una estructura más flexible en su uso para la interacción de las frases de una persona con un sistema de computo que implemente una tarea específica de un campo de estudio.

En este trabajo de graduación se verán algunos métodos que facilitan la traducción de textos, así como las bases que fundamentan la construcción de un sistema de esta magnitud, por medio de un prototipo que muestre la funcionalidad del sistema deseado.

1 MARCO TEÓRICO

1.1 Introducción

Una de las metas que se plantea dentro de la Inteligencia Artificial, es que las máquinas piensen y realicen tareas casi como si fueran humanas. Para que este objetivo planteado dentro de la Inteligencia Artificial sea cumplido, sería necesario implementar varias tareas que son propias del ser humano, como por ejemplo:

1. Reconocimiento de lenguaje natural.
2. Representación del conocimiento.
3. Razonamiento automatizado.
4. Aprendizaje

Existen además de éstas, otras tantas tareas que son realizadas por el ser humano, pero para este trabajo el objetivo central será el Reconocimiento de lenguaje natural y más específicamente aún su subdivisión en la rama de *traducción automática*.

1.2 Antecedentes

Es necesario partir de una base para poder construir un sistema experto para reconocimiento y traducción del lenguaje, es por ello que se plantean un grupo de antecedentes que enmarquen la factibilidad del proyecto.

1.2.1 Test de Turing

En 1950 Allan Turing desarrollo un proceso a través del cual proponía identificar la existencia de inteligencia en una máquina. A pesar de la lejanía en años de esta prueba sigue siendo una de las bases principales que sustentan a la inteligencia artificial hoy en día.

El fundamento principal de esta prueba, es que, si una máquina se comporta en todos los aspectos como inteligente, debe entonces ser inteligente.

La prueba consiste en un desafío. Se tiene un juez, quien conversará desde un teletipo con la máquina y un humano, estos últimos dos deben convencer al juez de quien es el humano en la conversación. Si el juez no es capaz de identificar quien es el humano entonces la máquina habrá pasado la prueba.

Allan Turing creía que para el año 2000 existirían máquinas lo suficientemente inteligentes que pasarían el test con un 30% de éxito.

1.2.2 A.L.I.C.E

Quizá este programa de diálogo entre una máquina y un humano sea el ejemplo más notable de inteligencia en una máquina. A. L. I. C. E, ha recibido 3 premios Loebner como la computadora más humana, el más reciente de ellos el 19 de septiembre de 2004.

Este programa de software consiste en la comunicación entre una persona y la máquina, simulando una conversación. Este es un sistema tanto de aprendizaje como de reconocimiento de lenguaje natural, ya que el sistema debe interpretar lo que el interlocutor humano dice y aprender de las frases del mismo.

1.3 Reconocimiento de Lenguaje Natural

1.3.1 Lenguaje Natural

El lenguaje natural es el medio por el cual el ser humano se comunica normalmente con otras personas.

El lenguaje natural es el que permite nombrar los objetos que rodean al ser humano y generar conceptos razonables acerca de ellos, el lenguaje natural nace y se desarrolla a partir de la experiencia humana y puede ser utilizado para analizar situaciones altamente complejas y razonar detalladamente. Los componentes semánticos (ligado al significado), da a los lenguajes naturales una alta capacidad de expresión y valor como una herramienta para razonamiento detallado. La sintaxis de un Lenguaje Natural puede ser modelada fácilmente por un lenguaje formal, similar a los que se utilizan en matemáticas y lógica, es en este punto donde el proyecto se enfocará principalmente ya que más que un proyecto para interpretación de significado (aunque si debe realizar análisis semántico) tiene una orientación más fuerte hacia el análisis del orden del lenguaje (análisis sintáctico), como se verá más adelante. Otra propiedad de los lenguajes naturales es la polisemántica, o sea, la posibilidad de que una palabra en una oración tenga diversos significados.

1.3.2 Lenguajes formales

A pesar de la importancia fundamental que tiene el lenguaje natural para los seres humanos, el lenguaje natural parece ser *inadecuado* para algunos casos, ya que el lenguaje natural contiene ambigüedades e imprecisiones que hacen que sea inadecuado analizarlo en algunos ámbitos (como para una máquina).

Es la necesidad de interpretar de manera efectiva el lenguaje natural, la que ha llevado al ser humano a construir *lenguajes artificiales*, entre los que destacan los *lenguajes formales* y los *lenguajes de programación*, esta sección se enfocará en el lenguaje formal.

El lenguaje formal es aquel que el hombre ha desarrollado para expresar las situaciones que se dan específicamente en algún área del conocimiento. Las palabras y oraciones que forman un lenguaje formal se encuentran claramente definidas, es decir, no existen ambigüedades.

Los lenguajes formales se encuentran libres de cualquier componente semántico fuera de sus operadores y relaciones. La principal ventaja, entonces, que presentan los lenguajes formales, es que en estos la ambigüedad es eliminada.

1.3.2.1 Características de los lenguajes Formales

- Los lenguajes formales son desarrollados a partir de una teoría previamente establecida, la cual se verá más adelante en este documento (ver. Traducción automática).
- El componente semántico utilizado es mínimo, es decir se pone mayor énfasis en el ordenamiento lógico de las palabras (componentes sintácticos).

- Presentan la posibilidad de incrementar el componente semántico de acuerdo con la teoría a formalizar.
- La sintaxis produce oraciones que no son ambiguas.
- El potencial de la construcción computacional es elevado debido a la completa formalización.

1.3.3 Aplicaciones del Reconocimiento de Lenguaje Natural

Las aplicaciones que se pueden encontrar en el área del Reconocimiento de Lenguaje Natural (RLN, de aquí en adelante) son muy variadas, el reconocimiento del lenguaje natural es por sí un campo de un alcance amplio, algunas de las aplicaciones de esta rama de la Inteligencia Artificial son:

1. *Traducción automática*: se refiere más que nada a la traducción correcta de un lenguaje a otro, tomando en cuenta lo que se quiere expresar en cada oración, y no solo palabra por palabra. Una aproximación a este tipo de traductores es el *babylon*.
2. *Recuperación de la información*: en esta aplicación, un claro ejemplo sería el siguiente: Una persona llega a la computadora y le dice, en lenguaje natural, que es lo que busca, la computadora busca y le dice que es lo que tiene referente al tema.
3. *Extracción de Información y Resúmenes*: Los nuevos programas, deben tener la capacidad de crear un resumen de un documento basándose en los datos proporcionados, realizando un análisis detallado del contenido y no solo la truncando las primeras líneas de los párrafos.
4. *Resolución cooperativa de problemas*: La computadora debe tener la capacidad de cooperar con los humanos para la solución de problemas complejos, proporcionando datos e información, incluyendo también, la demanda de información por parte del ordenador al usuario, debiendo existir una excelente interactividad entre el usuario y el ordenador.

5. *Tutores inteligentes*: La aplicación del PLN en este aspecto, viene siendo más académico, ya que se refiere a la enseñanza asistida por computadora, debiendo esta ser aprox. en un 99%, al tener esta la capacidad de evaluar al educando y tener la capacidad de adaptarse a cada tipo de alumno.
6. *Reconocimiento de Voz*: Esta es una aplicación del PLN que más éxito ha obtenido en la actualidad, ya que las computadoras de hoy ya tienen esta característica, el reconocimiento de voz puede tener dos posibles usos: para identificar al usuario o para procesar lo que el usuario dicte, existiendo ya programas comerciales, que son accesibles por la mayoría de los usuarios, ejemplo: ViaVoice¹.

Para este trabajo de graduación, como se puede observar las posibilidades que ofrece el RLN son amplias, sin embargo este trabajo se enfocará principalmente en una de estas aplicaciones para la construcción del Sistema Experto que permita un RLN exitoso, de aquí en adelante, la metodología (aplicación) que se utilizará para la construcción del Sistema Experto de RLN, será la Traducción Automática.

1.4 Traducción Automática

En su sentido más general la traducción Automática (TA de ahora en adelante), se refiere a la transferencia de pensamiento y conceptos de una lengua fuente hacia una lengua destino (meta). La traducción no se trata de únicamente realizar una búsqueda de palabras en un diccionario, más bien consiste en la transmisión del pensamiento del ser humano hacia una lengua distinta.

¹ <http://www-306.ibm.com/software/voice/viavoice/>

Si bien la TA es una aplicación del Reconocimiento de Lenguaje Natural, que tiene por finalidad transmitir los pensamientos de una lengua a otra lengua humana, debido a las metodologías ofrecidas por la TA se ha encontrado que para este trabajo los conceptos relacionados a la TA pueden igualmente ser aplicados para la traducción de un lenguaje natural (idioma español), hacia un lenguaje de conceptos (marcos, por ejemplo). A continuación se verán las metodologías ofrecidas por la TA y los conceptos que serán utilizados para la construcción del Sistema Experto de este trabajo de graduación.

1.4.1 Metodologías básicas utilizadas

Es necesario conocer cuáles son las necesidades básicas de un sistema típico de traducción automática, así como las estrategias a seguir para lograr que un programa traduzca expresiones en lenguaje natural.

Un traductor humano usa al menos cinco tipos distintos de conocimiento²:

- Conocimiento de la lengua origen (LO)
- Conocimiento de la lengua meta (LM)
- Conocimiento de las distintas correspondencias entre la LO y LM. En el más básico de los niveles, es el conocimiento del equivalente de traducción de cada palabra
- Conocimiento del dominio sobre el que se está traduciendo, incluyendo conocimientos generales y de *sentido común*.
- Conocimiento del entorno cultural de los hablantes de LO y LM

² Arnold, D., L. Balkan, S. Meijer, R. Lee Humphreys, & L. Sadler (1994) *Machine Translation: An Introductory Guide*. London: Blackwell-NCC

Tradicionalmente se han distinguido varios niveles de conocimiento lingüístico:

- Conocimiento fonológico: conocimiento sobre el sistema de sonidos de un lenguaje. Por lo que respecta a textos escritos, este tipo de conocimiento no es realmente relevante. Sí lo es para aplicaciones de traducción automática que incluyan soporte para reconocimiento de voz. En cualquier caso, sí es crítico para cualquier sistema el conocimiento ortográfico.
- Conocimiento morfológico: conocimiento de la manera en que las formas de un determinado lema son construidas, así como de las posibles producciones de una determinada raíz.
- Conocimiento sintáctico: cómo las distintas palabras se combinan de forma lineal para construir frases y oraciones aceptables de una lengua. Por lo que respecta al tratamiento automático de una lengua es, básicamente, un problema de ordenación de cadenas de caracteres.
- Conocimiento semántico: el conocimiento del significado de palabras y frases y de las relaciones de significado entre una frase y sus unidades constituyentes.

Por si solo, el conocimiento morfológico no tendría sentido para la interpretación de la LO (idioma español), para el Sistema Experto a construirse, en cambio la utilización del conocimiento sintáctico aunado al anterior tiene mayor relevancia y significado para el objetivo de este trabajo de graduación, aún cuando es importante el conocimiento semántico para fines prácticos de este trabajo no se abarcará en un alto grado, debiendo ser realizada la interpretación de los componentes léxicos y sintácticos para la LM, por un Sistema Experto para tal efecto, el cual no será desarrollado en este trabajo.

1.4.2 El análisis sintáctico

De todos los niveles de análisis (léxico, semántico, sintáctico), la sintaxis ha sido durante mucho tiempo el análisis al que la lingüística ha prestado mayor atención. En cuanto al tratamiento automático del lenguaje natural las razones de tal importancia son las siguientes³:

- El procesamiento semántico funciona sobre los constituyentes de la oración. Si no existe un paso de análisis sintáctico, el sistema semántico debe identificar sus propios constituyentes. Por otro lado, si se realiza un análisis sintáctico, se restringe enormemente el número de constituyentes a considerar por el semántico, mucho más complejo y menos fiable. El análisis sintáctico es mucho menos costoso en términos computacionales, que el análisis semántico (que requiere inferencias importantes). Por tanto, la existencia de un análisis sintáctico conlleva un considerable ahorro de recursos y una disminución de la complejidad del sistema.
- Aunque frecuentemente se puede extraer el significado de una oración sin usar hechos gramaticales, no siempre es posible hacerlo.

Como puede observar, es entonces necesario realizar un análisis sintáctico previo a una interpretación en un nivel superior (debe de recordarse que esta interpretación debe ser realizada por un Sistema Experto especializado en esta tarea).

³ Antonio Moreno Ortiz, *“lingüística, fonética, fonología, lenguaje, traducción”*. <http://elies.rediris.es/elies9/3-1-2.htm>, consultado Agosto de 2004.

La sintaxis contempla dos modalidades de análisis. El *análisis de constituyentes*, que no es más que el análisis de la estructura de la frase, es decir, la estructuración de las oraciones en sus partes constituyentes y la categorización de éstas como nominales (verbos, adjetivos). El *análisis de las relaciones*, o también análisis de *funciones gramaticales*, que es la asignación de relaciones gramaticales como sujeto y objeto, por ejemplo.

Para terminar este apartado, es importante definir que, formalmente, una *lengua* es un conjunto de *oraciones*, en donde una oración es una cadena de uno o más símbolos que pertenecen al vocabulario de la lengua. Entonces como resultado se obtiene una gramática que no es más que una especificación formal y finita de un conjunto de oraciones.

1.4.3 Sistemas basados en Conocimiento

La traducción basada en Conocimiento (KBMT: *Knowledge-Based Machine Translation*, de aquí en adelante) aboga por la creación y utilización de un amplio depósito de información en el proceso de traducción, no solamente información lingüística, sino también sobre conocimiento del mundo o dominio específico de traducción. Esta podría ser considerada como la característica fundamental, con este tipo de enfoque, para la construcción del sistema que interesa, se persigue la creación de una extensa y bien estructurada base de conocimiento.

La premisa básica de la KBMT es que para obtener una traducción de calidad y precisión es necesaria una completa interpretación semántica del texto origen (esta interpretación semántica será realizada por el Sistema Experto que analice las propuestas de ley). Por lo tanto, los componentes de

análisis y generación de un sistema de KBMT deben poseer al menos los siguientes componentes:

- Una gramática para la lengua
- Un léxico para la lengua
- Un conjunto de conceptos para el dominio
- Un conjunto de reglas que proyecten estructuras sintácticas sobre estructuras semánticas para el análisis
- Un conjunto de reglas que proyecten estructuras semánticas sobre estructuras sintácticas para la generación

Como característica distintiva del KBMT, se pretende utilizar un *conocimiento* del dominio que subyace a la traducción, lo cual implica la existencia de una base de conocimiento que sirva de soporte para la ontología de conceptos. Esta base de conocimiento debe ser utilizada en las posteriores etapas de análisis del lenguaje.

1.5 Conclusiones

La construcción del sistema experto de reconocimiento de lenguaje se llevará a cabo en base a los métodos y aplicaciones presentadas en este marco teórico. Como se puede observar en este marco teórico, es necesaria la representación del conocimiento, sobre todo del conocimiento natural, es por ello que debe realizarse una base de hechos, esta base de hechos será realizada a través de marcos.

No es suficiente tener la información almacenada en una base de hechos, sin embargo el proceso de construcción de este sistema Experto de Reconocimiento de Lenguaje Natural utilizará esta base de hechos (diccionario) como base fundamental para el análisis que es necesario realizar en el

reconocimiento del lenguaje. Por lo que se debe analizar los componentes léxicos del lenguaje y su estructura (análisis sintáctico).

En la construcción de este SE debe ser tomada en cuenta la estructura del lenguaje de entrada y la generación de conocimiento para la alimentación de un Sistema Experto Legislativo, para ello se debe representar primeramente el conocimiento común a ambos Sistemas y el conocimiento relativo al lenguaje natural, es decir estructura gramatical, léxico, etc.

2 ENTRADAS DEL SISTEMA

2.1 Análisis

Antes de definir el diseño de cualquier sistema, es necesario conocer los requerimientos funcionales con los que debe cumplir el mismo y hasta entonces solucionar dichos requerimientos. Es por ello que previo al diseño de este proyecto, se definirán los requerimientos funcionales con los que debe cumplir el mismo, a fin de que el lector pueda identificar las bases sobre las cuales se edificará éste proyecto.

2.1.1 Requerimientos Funcionales

Una tarea previa a la identificación de los requerimientos funcionales, los cuales son las funciones y procesos que debe realizar un sistema, consiste en la investigación del problema y no la forma en la que deberá definirse la solución, esta tarea ya fue desarrollada en el capítulo anterior en donde se definió que el problema consiste en realizar un sistema de reconocimiento de lenguaje natural, con una fuerte orientación hacia el reconocimiento de términos comúnmente utilizados en la Constitución de la República de Guatemala, este sistema deberá “traducir” el lenguaje humano (más específicamente el idioma español), hacia un lenguaje meta (LM) que sea útil para que un Sistema Experto Legislativo pueda realizar un análisis del significado de lo que el humano exprese a través de este sistema de reconocimiento de lenguaje natural.

Para que el sistema pueda cumplir con su cometido debe llenar entonces con los siguientes requerimientos funcionales:

- Reconocimiento de una lengua origen, que será el idioma español (este reconocimiento se delimitará en el siguiente apartado de este capítulo).
- Generación de una lengua meta, ésta proporciona las entradas para el funcionamiento de un Sistema Experto Legislativo.
- Creación de nuevo conocimiento en el sistema, éste lo deberá proporcionar el usuario común del sistema, el sistema debe proporcionar los “conceptos” fundamentales que utiliza, para que el usuario agregue su conocimiento acerca de estos conceptos (este punto se ampliará en el capítulo 4).

2.1.2 Límites

La funcionalidad y procesos que pueden realizar los sistemas de reconocimiento de lenguaje abarca un campo muy amplio como ya se mostró en el capítulo anterior, es por ello necesario definir el área que se abarcará con este proyecto de graduación, es decir establecer límites, a continuación se definen límites del conocimiento del sistema y estructura.

2.1.2.1 Conocimiento

Al igual que un niño, este sistema contará con información proporcionada por su creador en una base de conocimientos (diccionario), sin embargo es necesario que la información en el sistema se incremente, a fin de que el conocimiento del mismo sea satisfactorio para las necesidades de información de sus usuarios, por lo que se debe proporcionar una interfaz por medio de la cual cualquier usuario pueda incrementar el contenido del diccionario del sistema, tomando como base los conceptos básicos con los que contará el mismo.

La fuente principal de conocimiento para este sistema de reconocimiento de lenguaje proviene de los artículos 1 al 10 de la Constitución de la República de Guatemala, este conocimiento será esencial para la generación de la lengua meta.

2.1.2.2 Estructura

Debido a que el lenguaje humano es un conjunto de oraciones que se entrelazan entre sí para comunicar un mensaje completo, el usuario debe comunicar sus ideas al sistema en forma de oraciones completas y estructuradas para que éste pueda analizar la información del lenguaje origen proporcionada por sus usuarios.

Este sistema no se encargará del análisis semántico en un sentido estricto de lo que este análisis implica en el lenguaje, este análisis lo llevará a cabo el sistema Experto Legislativo, con el que éste sistema de reconocimiento de lenguaje ha de interactuar.

Entonces este sistema debe realizar un profundo análisis léxico y sintáctico de las frases utilizadas por el usuario, y en menor grado, análisis semántico para realizar la identificación de conceptos utilizados por el Sistema Experto Legislativo dentro de las oraciones proporcionadas por el usuario.

2.1.3 Alcances

El sistema cuenta con dos módulos principales. El primero es el módulo de comunicación con el usuario, por medio del cual el usuario puede ingresar la propuesta de ley que le interesa por medio de oraciones simples, este módulo es el medio de comunicación entre el usuario y el Sistema Experto Legislativo, ya que es a partir de éste que se podrá realizar la traducción de la lengua origen, que es la transformación de las oraciones formadas por el usuario hacia la lengua meta, que son los conceptos reconocidos por el Sistema Experto Legislativo.

Junto con éste sistema se proporcionará al usuario la posibilidad de incrementar el conocimiento del mismo, esta tarea se realiza por medio de un sistema alterno, que en realidad constituye un módulo de este Sistema de Reconocimiento, por medio del cual se muestran al usuario, los conceptos con los que cuenta el mismo y así el usuario podrá crear nuevas instancias relacionadas con el conocimiento que forma parte del Sistema de Reconocimiento.

Para el correcto funcionamiento de un sistema es necesario que exista una acción para la cual se genere una respuesta correspondiente, es decir, que exista un conjunto de acciones válidas que al interactuar con un sistema produzca de éste una respuesta válida.

Para este Sistema de Reconocimiento de Lenguaje Natural, el conjunto de acciones válidas por medio de las cuales se puede interactuar con el sistema son un conjunto de frases en lenguaje natural, es decir idioma español, que han

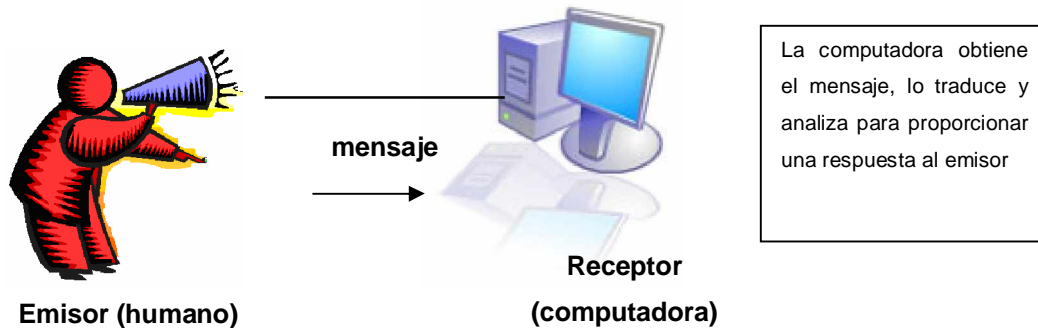
de ser proporcionadas por el usuario por medio de una interfaz gráfica en la que pueda expresar sus ideas (iniciativa de ley).

Este proceso de ingreso de una lengua origen (LO), conlleva posteriormente a un proceso de análisis léxico y sintáctico, los cuales son definidos más adelante en este capítulo. Con este proceso se obtiene como respuesta la obtención de una lengua meta (LM), la cual sirve como alimentación de cualquier otro sistema experto (específicamente en el caso de este trabajo, un sistema experto legislativo). Esta lengua meta se define por medio de estructuras definidas, las cuales se analizarán a detalle en el capítulo 5 de este trabajo.

2.1.4 Lengua Origen (LO)

El lenguaje natural, específicamente el idioma español tal y como se conoce, consta de un conjunto de frases que juntas tratan de transmitir una idea a quien recibe la información (receptor), por parte de quien las produce (emisor) así que se puede ver que en toda comunicación entre seres humanos es necesaria la existencia de alguien quien produce la información (emisor) y alguien que la recibe y la transforma para uso de la misma (receptor). Tal es el caso de este Sistema de Reconocimiento de Lenguaje Natural quien se convierte en el receptor de la información brindada por el emisor humano, quien a través de frases transmite la idea que desea comunicar (Lengua Origen, LO), como se ha especificado al inicio de este capítulo, las ideas que debe brindar el transmisor a este sistema, deben estar relacionadas con el campo del derecho, limitándose para ello a conceptos básicos que se pueden encontrar en la Constitución de la República de Guatemala en sus artículos 1 al 14.

Figura 1. Proceso de comunicación



2.1.5 Conceptos del Lenguaje Natural

El lenguaje natural, visto desde el punto de vista de este proyecto de graduación, de ahora en adelante se definirá como una o varias oraciones o frases que en su totalidad engloban el mensaje que se desea transmitir para que el Sistema de Reconocimiento de Lenguaje Natural lo traduzca, esto debido a que la oración es la menor unidad del habla humana, la cual comunica un sentido completo.

Desde este punto de vista se puede entonces identificar conceptos esenciales de la estructura gramatical de la oración. Es importante anotar que las oraciones, cualesquiera estas sean, están formadas por dos partes significativas que se denominan **sintagmas**, a mayor detalle, a estos se les conoce como sujeto y predicado, se dice que estas son significativas porque están dotadas de significado. Por ejemplo, en la oración: *La persona tiene derechos y obligaciones*, se dice que *La persona* es la parte de la oración que indica al sujeto, de la que se dice o expresa una acción que realiza; *tiene derechos y obligaciones* indica al predicado que es lo que se predica o dice del sujeto.

A su vez si se adentra con mayor detalle hacia la estructura de los sintagmas, se podría observar que estos constan de elementos básicos que les identifican, estos son los núcleos del sujeto y del predicado. En este orden el sujeto tiene una palabra principal que constituye el núcleo o palabra alrededor de la cual giran otras palabras que complementan su sentido, a esta palabra se le denomina sustantivo. El núcleo del predicado tiene una palabra principal que constituye su núcleo, este es el verbo, los verbos designan todos los eventos, acciones, procesos y fenómenos que ocurren en la realidad.

2.1.5.1 Clasificación de las palabras

En el idioma español pueden identificarse las palabras por diferentes tipos, como se vio anteriormente, la oración esta dividida en sintagmas, sin embargo estos a su vez están conformados por diferentes componentes del idioma por medio del cual pueden ser clasificadas las palabras. Esta clasificación es importante para este Sistema a fin de conocer los elementos que se colocarán en la base de hechos, con un mayor detalle de visión, a continuación se listan los diferentes tipos de palabras:

1. **Sustantivos:** Estas son las palabras que nombran a las personas, animales, cosas sentimientos y situaciones
2. **Verbos:** son las palabras que designan las acciones que realiza el sustantivo.
3. **Adjetivos:** son palabras que mencionan las cualidades de los sustantivos.
4. **Adverbios:** son palabras que determinan y califican a los verbos

Tipo de Adverbio	Adverbios
De lugar	Aquí, allí, cerca, lejos
De modo	Despacio, rápido, claramente, fácil
De tiempo	Hoy, ayer, antes, temprano, mañana
De cantidad	Más, poco, mucho, tanto
De orden	Últimamente, sucesivamente
De afirmación	También, sí, claro
De negación	No, nunca, jamás
De duda	Quizá, tal vez, acaso

Tabla 1. Tipos de Adverbios

5. **Determinantes:** son palabras que especifican datos del sustantivo. Los determinantes se clasifican en *artículos* y *Determinantes*.

Tipo de artículo	Artículo
Definidos	El, lo, la, los, las
Indefinidos	Un, unos, alguien, algo, ...

Tabla 2. Tipos de Artículos

Tipo de determinantes	Determinante
Posesivos	Mío, mía, nuestros, ...
Demostrativos	Ése, éste, aquello, ...
Numerales	Primero, uno, vigésimo quinto, ...

Tabla 3. Tipos de determinantes

6. **Las preposiciones y las conjunciones:** Son palabras que sirven para relacionar otras palabras o ideas. Las preposiciones son: a, ante, bajo, cabe, con, contra, de, desde, en, entre, hacia, hasta. Las conjunciones

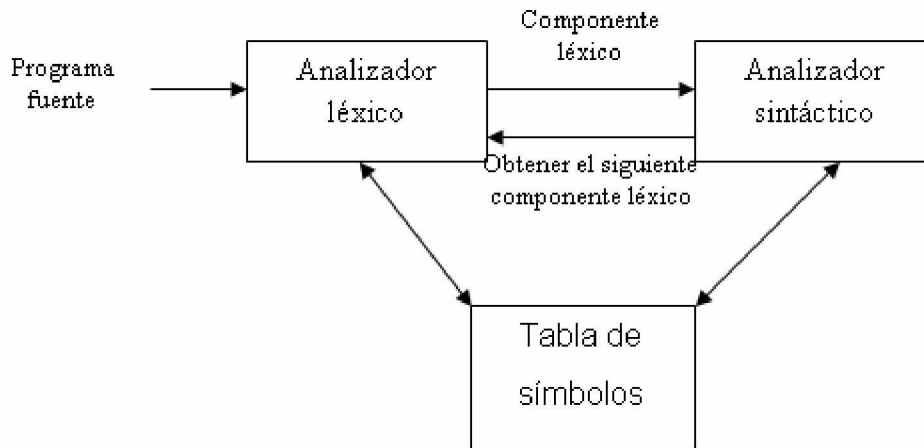
son: y, o, pero, aunque, ya, sin embargo, no obstante, por tanto, después, porque, etc.

2.2 Proceso de Reconocimiento

El objetivo principal que se persigue en cualquier Sistema de Reconocimiento de Lenguaje natural, es el reconocimiento del lenguaje humano y que el Sistema pueda procesar el lenguaje, reconocerlo y posteriormente ofrecer una respuesta o acción correspondiente al mensaje del humano. En este apartado se estudia el proceso de reconocimiento que debe de llevar a cabo el sistema, y el análisis correspondiente que necesita realizar para tal efecto.

2.2.1 Análisis Léxico

Desde el punto de vista de los compiladores el análisis léxico consiste en la lectura de caracteres de entrada en el compilador, para elaborar como salida una secuencia de componentes léxicos que posteriormente han de ser utilizados por el analizador sintáctico para realizar el análisis correspondiente, como se puede apreciar en la figura 2.

Figura 2. Proceso del Análisis Léxico

Como ya se mencionó, los compiladores requieren en su fase de análisis léxico de la lectura de caracteres de entrada, para el caso de este sistema no es necesario realizar este análisis a ese nivel de detalle, ya que los componentes léxicos se encuentran almacenados en un lexicón, un lexicón es un diccionario, por así decirlo, que se coloca en la base de hechos del sistema, el proceso de análisis léxico consiste en identificar dentro de los patrones ingresados por el usuario en el Sistema, a aquellos que se equiparen con los componentes léxicos que existen en el lexicón. Dentro de los componentes léxicos que ingrese el usuario del sistema únicamente deben de ser descartados caracteres tales como espacios en blanco, comas y signos de puntuación. Teniendo este conocimiento, se podría entonces equiparar aquellos símbolos que el usuario haya ingresado dentro de su frase o frases. Entonces el análisis léxico consiste en la lectura del texto fuente y paso de componentes léxicos al analizador sintáctico como tareas principales, y una función secundaria en la interfaz del usuario consiste en la eliminación de la lengua origen de caracteres de espacios en blanco, tabuladores, símbolos de puntuación (coma, punto, punto y coma, ...) y caracteres de nueva línea.

En el siguiente apartado de este capítulo se estudiará el análisis sintáctico que se realizará en este sistema, pero antes de ello véanse los siguientes conceptos. Cuando se menciona el análisis sintáctico, se emplean con significados específicos los términos *componente léxico*, *patrón* y *lexema*. En general, hay un conjunto de cadenas en la entrada para el cual se produce como salida el mismo componente léxico. Este conjunto de cadenas es descrito mediante una regla denominada *patrón*, el cual se asocia al componente léxico. Se dice entonces, que el patrón concuerda con cada cadena del conjunto. Un *lexema* es una secuencia de caracteres en la lengua origen con la que concuerda el patrón para un componente léxico.

Debe de tomarse en cuenta que los componentes léxicos que reconocerá el Sistema de Reconocimiento ya fueron definidos en este mismo capítulo en el apartado **2.1.5.1** (*Clasificación de las palabras*). Los lexemas válidos que reconocerá el sistema y que formarán parte de su lexicón, serán definidos en el siguiente capítulo, como parte de la elaboración del Sistema.

2.2.2 Análisis Sintáctico

Al hablar de sintaxis y análisis sintáctico, se entiende estos que tratan la estructura de los lenguajes, es decir, la sintaxis muestra el rol de las palabras en una oración, en el caso de este trabajo de graduación se refiere a la estructura del idioma español.

Luego de realizado el análisis léxico, el analizador sintáctico obtiene una cadena de componentes léxicos del analizador léxico, y se comprueba si la cadena puede ser generada por la gramática del lenguaje origen.

A fin de reconocer entonces la estructura de los componentes léxicos brindados por el analizador léxico es entonces necesario que se ordenen a

estos de manera que ofrezcan un ordenamiento lógico, se sabe que en el idioma español las oraciones tienen una estructura predefinida, la cual da sentido a las palabra que expresa el ser humano, y esta estructura consiste claramente de un sujeto y predicado, pero persiste el problema de representación de este conocimiento, una forma efectiva de representarlo es por medio de una gramática, una gramática consta de símbolos terminales y no terminales, en donde los símbolos terminales de este sistema son los elementos léxicos incluidos en el lexicón, obsérvese como se puede definir una gramática sencilla para el sistema:

$o \rightarrow s p$

$s \rightarrow det n$

$s \rightarrow det n adj$

$p \rightarrow v s$

En donde o es la producción de la oración, constituida por un sujeto (s) y un predicado (p), det es el determinante de la oración, n indica el nombre o sustantivo, adj es el adjetivo y v es el núcleo, en esta gramática se puede identificar a los símbolos terminales en det , n , adj y v . En el siguiente capítulo se amplía esta misma gramática para todas las clases de palabras que ya fueron previamente definidas.

Es entonces por medio de esta gramática que se proporcionará la información necesaria para el Sistema Experto Legislativo con el que debe interactuar, equiparando las palabras analizadas con aquellos conceptos que han sido definidos para el correcto funcionamiento del Sistema Experto Legislativo.

Ahora se muestra un ejemplo de implementación en un sistema de Reconocimiento par el cual no se implementa el análisis sintáctico y para así, poder observar la importancia que proporciona el análisis sintáctico en un Reconocedor de Lenguaje Natural.

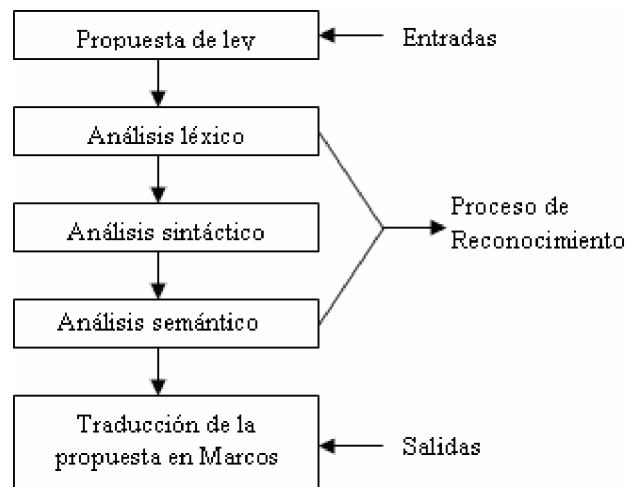
2.2.2.1 Emparejamiento de Patrones (Eliza)

Este método es utilizado por el programa Eliza, utiliza una base de datos de pares de entrada y salida, en donde la parte de entrada es una plantilla que debe ser emparejada contra la entrada del usuario y luego de que una pareja es seleccionada se presenta en la salida una respuesta. Una ventaja que brinda este método es la flexibilidad que ofrece, por ejemplo si se coloca en la entrada: *X computadores Y*, entonces se obtendría como respuesta: *Te interesan las computadoras?*. Como se puede ver otra ventaja de este método es su facilidad de ser programado, sin embargo, con este método no es conocido nada con respecto a la estructura (sintaxis), si en la entrada se obtuviera: *Yo te X*, se podría obtener como respuesta: *Porqué tu me X?*, siendo de esta manera correcto, sin embargo si se tiene: *Yo no te X*, de cualquier manera la respuesta sería: *Porqué tu me X?*, siendo de esta manera inconsistente su estructura. Otro problema que surge es el desconocimiento del significado (semántica), por ejemplo, en la entrada se obtiene: *Me siento X*, y la respuesta sería: *Siento mucho que te sientas X*, esta respuesta estaría bien si X fuera deprimido, pero no es así si por ejemplo X es feliz. En conclusión para la correcta comprensión del lenguaje natural son necesarias tanto la sintaxis como la semántica.

3 SALIDAS DEL SISTEMA

Todo sistema debe de desempeñar una tarea que produzca un conjunto de resultados esperados, sin embargo para ello son necesario pasos previos que colaboran en la consecución de un objetivo, que para el caso de este trabajo de este trabajo de graduación, consisten en la producción de un lenguaje destino.

Figura 3. Análisis Sintáctico



En la figura 3 podemos observar el esquema del proceso de análisis sintáctico, en el caso de este trabajo consiste de las siguientes tareas:

- Propuesta de ley: las entradas para el inicio del proceso de análisis sintáctico, son formadas por las propuestas de ley que ingresa un experto en forma de oraciones.
- Análisis léxico: como fue explicado en la sección 2.2.1, este proceso consiste en la lectura de caracteres de entrada, de tal manera que se pueda elaborar una secuencia de componentes léxicos, que puedan ser reconocidos durante el proceso de análisis sintáctico.

- Análisis sintáctico: debe de realizarse un análisis que permita verificar la validez en el orden de cualquier frase de entrada.
- Análisis semántico: el orden en una oración no es suficiente para realizar un conjunto de salidas básicas, es necesario realizar un análisis de las frases de tal manera que se obtenga el significado de cada componente léxico, colocado de manera correcta, al hablar de sintaxis.
- Traducción de la propuesta en marcos: en este punto del flujo de tareas que se ilustra en la figura 3, es en donde se obtiene el final de las tareas y las salidas esperadas, que son las oraciones traducidas en su equivalente en marcos.

El reconocimiento de verbos se llevará a cabo durante el análisis léxico. Debe de tomarse en consideración que existen verbos regulares e irregulares, siendo los verbos regulares similares en su conjugación, en tanto que los verbos irregulares son distintos en cuanto a su conjugación en los diferentes tiempos y personas.

El reconocimiento de verbos regulares consiste en el reconocimiento de patrones que darán como resultado el verbo en infinitivo para obtener como resultado un hecho el verbo en infinitivo. Por ejemplo:

Verbo: *amar*

Amaré

Amé

Amo

Amaré

Amarás

Como se puede observar en el ejemplo, se ha sombreado la parte en común del verbo que se ha conjugado, lo cual dará como resultado un emparejamiento de patrones de letras.

En cuanto a los verbos irregulares, es importante hacer notar que aunque la conjugación de los verbos no observa un patrón que sea fácilmente reconocible, sin embargo en este sistema el campo puede ser delimitado, ya que las propuestas de ley son formuladas en tercera persona singular y plural (el, ella, ellos, ellas),

Por ejemplo, en artículo No. 3 de la constitución se enuncia: *El Estado garantiza y protege la vida humana...* En esta frase el sujeto es *Estado*, y los verbos en infinitivo son *garantizar* y *proteger*, como se puede ver este es un ejemplo de una ley y por supuesto las propuestas de ley son formuladas de similar manera, lo cual delimita el accionar del Sistema de Reconocimiento, al ser formuladas las propuestas de ley de manera impersonal. Esto conduce entonces, a definir la forma en la cual serán reconocidos los verbos irregulares, para al igual que los verbos regulares, pueda obtenerse una salida de los mismos en infinitivo. Luego de lo anterior se añadirá entonces que los verbos irregulares tendrán únicamente tres conjugaciones, presente, pasado y futuro, en tercera persona para un reconocimiento más sencillo de los mismos.

3.1 Interpretación de Oraciones

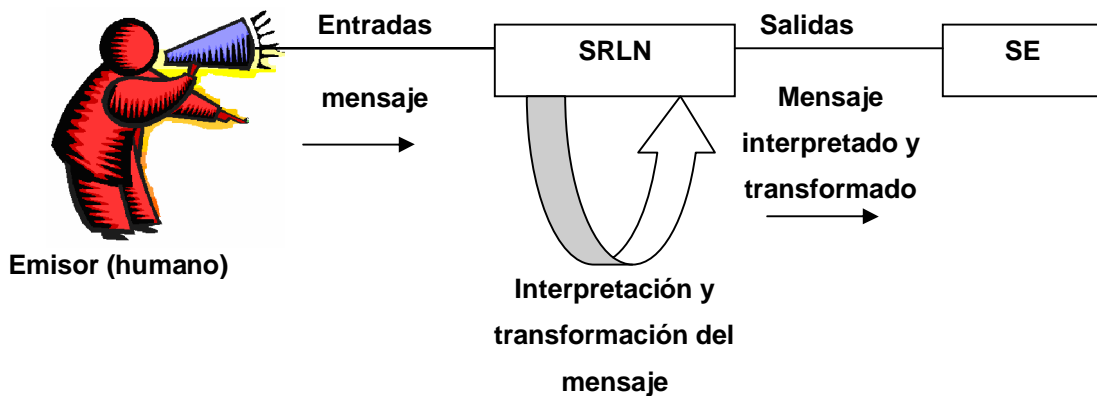
El principal objetivo del Sistema de Reconocimiento de Lenguaje Natural (SRLN) de este trabajo consiste en proveer de información a otros sistemas, específicamente a un Sistema Experto Legislativo, de tal manera que este sistema pueda obtener la información para su análisis en un formato que sea fácil de interpretar y proporcione un orden lógico de las palabras para la

comprensión del sistema destino, es decir, se debe definir una gramática para el mismo.

Como se explicó en capítulos anteriores, en toda comunicación existe un emisor y un receptor, en el caso de las salidas que proporciona este SRLN⁴, es la forma de comunicación que existe entre dos sistemas diferentes, siendo el SRLN el emisor y el Sistema Experto Legislativo (SE) constituye el receptor, el mensaje es la oración que proporciona el SRLN en el formato específico a ser interpretado por el SE.

Es entonces la oración interpretada por el SRLN el producto final esperado, para proporcionar las salidas deseadas. Como se muestra en la figura 4, el SRLN es el punto intermedio de comunicación entre el legislador y el SE.

Figura 4. Transformación de mensajes



⁴ SRLN, Sistema de Reconocimiento de Lenguaje Natural

En esta sección no es importante centrarnos en el proceso en sí, que conlleva la transformación de las frases del usuario humano, hacia la gramática utilizada por el SE, de cualquier manera es importante señalar que este proceso se lleva a cabo por medio de algunas de las metodologías abarcadas en los capítulos anteriores, como lo son el emparejamiento de patrones utilizado por Eliza (ver sección 2.2.2.1), a fin de reconocer palabras compuestas, que por sí solas no tienen el significado esperado, tal es el caso de “menor de edad” o “persona de la tercera edad”, para reconocer este tipo de palabras fue necesario el uso de reconocimiento de patrones, en otros casos como lo son los verbos regulares, igualmente fue utilizado el reconocimiento de patrones.

Para realizar tanto el análisis léxico de las entradas como el ordenamiento de la oración de salida, es decir análisis sintáctico, se lleva a cabo por medio del lenguaje de reglas. Y por último será mencionado el uso de diccionarios, es decir, el lexicón tan importante para este tipo de sistemas, sin el cual se vería afectado el funcionamiento exitoso del SRLN, siendo este lexicón la base fundamental para realizar las metodologías previamente descritas. En el siguiente capítulo, se adentrará al lector a un mayor detalle de la construcción del SRLN y su funcionamiento interno.

4 CONSTRUCCIÓN DEL SISTEMA

Hasta ahora se ha mostrado la forma en la que se espera que funcione el SRLN y la teoría que cimienta las bases para el mismo, sin embargo para poner en marcha el SRLN es necesario codificar y poner en práctica esta teoría. En este capítulo serán analizados los distintos componentes que fundamentan al SRLN y que unidos conforman un sistema completamente funcional.

4.1 Lexicón

El lexicón lo constituyen todas las palabras de la base de hechos que han sido incluidas en el SRLN, el lexicón esta formado por todos los tipos de palabras que se pueden encontrar en la estructura de una oración, como lo son, artículos, sustantivos, adjetivos, verbos, adverbios, preposiciones, etc. Es importante hacer notar que aún cuando el lexicón forma parte en realidad de la base de hechos, aquí se toma en cuenta como elementos separados e independientes uno del otro, a fin de hacer una clara distinción entre los elementos propios del lenguaje natural, es decir el lexicón, y aquellos elementos que son importantes para el control del proceso de reconocimiento en el sistema. Entonces, el lexicón se define como el conjunto de elementos que forman parte de las oraciones (palabras) y que cumplen un rol importante en el análisis léxico y posterior traducción que produce las salidas del sistema.

Previo a la definición de los elementos que conforman el lexicón, es importante hacer notar que a su vez, estos elementos se encuentran agrupados en diferentes tipos y a la vez todos son palabras. Para la definición de estos objetos que forman parte del lexicón, es necesario mostrar a la vez las clases que definen los objetos.

4.1.1 Palabra

Al hablar de clases, se puede pensar en la herencia, de una clase más general hacia clases especializadas que obtienen propiedades de la clase padre. Es así como la palabra es el elemento más atómico que se utiliza por el SRLN, al decir atómico, se entiende que es la unidad mínima de información, o lo que es lo mismo, la palabra es la clase padre de la cual las otras clases heredarán sus propiedades. A continuación se muestra la estructura de la clase PALABRA en pseudo-código:

```
(defclass PALABRA (is-a USER)
  (role abstract)
  (multislot palabra))
```

Esta clase no se puede acceder directamente para cambiar alguna de sus propiedades, es por ello que fue definida con un rol abstracto, es decir, esta es la representación de las palabras que más adelante se mostrarán. Su único atributo, es palabra, esta propiedad es heredada a todos aquellos elementos del lexicon que *son palabras*.

4.1.2 Artículo

Como se describió en el capítulo 2, los artículos son algunos de los tipos de palabras que existen, estos definen en términos generales el género de los sustantivos y su cantidad. Los artículos preceden entonces a los sustantivos, la estructura de estos utilizada por el SRLN es la siguiente:

```
(defclass ARTICULO (is-a PALABRA)
  (pattern-match reactive)
  (role concrete)
  (multislot palabra (create-accessor write))
  (slot tipo (create-accessor write)))
```

En la definición de la clase ARTÍCULO se define que un artículo es una palabra (*is-a PALABRA*), luego se define que esta debe de reaccionar al encontrarse una equiparación en una regla, en los elementos que componen al artículo se puede decir que existen dos, la palabra, es decir el artículo como tal y el tipo de artículo, ya sea definido o indefinido.

4.1.3 Sustantivo

Los sustantivos son aquellas palabras que describen a algún objeto, persona o algún elemento sobre el que se hace mención en la oración, el sustantivo puede presentarse tanto en el sujeto como en el predicado, siendo de cualquier manera la palabra afectada en la oración. La estructura que define a un sustantivo en el SRLN, es la siguiente:

```
(defclass SUSTANTIVO (is-a PALABRA)
  (pattern-match reactive)
  (role concrete)
  (slot palabra (create-accessor write)))
```

La clase sustantivo hereda de la clase PALABRA, su único atributo que es palabra, este atributo almacena todas las palabras que pueden identificarse como sustantivos.

4.1.4 Preposición

Las preposiciones sirven en la oración para encadenar ideas y relacionarlas a la vez, de manera general una preposición precede a una palabra, un sustantivo, a la cual afectará una acción correspondiente. La estructura que define a una preposición, es la siguiente:

```
(defclass PREPOSICION (is-a PALABRA)
  (pattern-match reactive)
```

(role concrete)
(multislot palabra (create-accessor write)))

4.1.5 Determinante

Los determinantes, aunque son un tipo de palabras, identifican a los sustantivos de una manera indirecta. La estructura que define a una preposición, es la siguiente:

```
(defclass DETERMINANTE (is-a PALABRA)
  (pattern-match reactive)
  (role concrete)
  (multislot palabra (create-accessor write))
  (slot tipo (create-accessor write)))
```

La clase sustantivo hereda de la clase PALABRA, el atributo palabra, este atributo almacena todas las palabras que pueden identificarse como determinantes, además en el slot tipo pueden ser almacenados dos valores que son *posesivo* y *demonstrativo*, los cuales corresponden a los tipos de determinantes que existen.

4.1.6 Verbo

La clase VERBO es un tipo de palabra, por lo cual hereda sus atributos de la clase PALABRA, sin embargo VERBO a su vez es una clase padre que hereda sus atributos a otras clases hijo, las cuales, como se podrá ver más adelante, son tipos de verbos que aun cuando su contexto cambie no dejan de ser verbos. La estructura que define a un verbo, es la siguiente:

```
(defclass VERBO (is-a PALABRA)
  (role abstract)
  (slot palabra))
```

La clase verbo no puede ser accedida directamente a sus atributos, esta es únicamente definida como una especie de plantilla para los distintos tipos de verbos existentes, es decir, será la clase padre, para las clases que definen los tipos de verbos que utiliza el SRLN.

4.1.7 VERBO_REGULAR

Los verbos pueden ser fácilmente clasificados, en una primera instancia, como regulares e irregulares, siendo los verbos regulares aquellos cuya conjugación no sufre cambios significativos con respecto al verbo en infinitivo. La estructura que define a un verbo regular, es la siguiente:

```
(defclass VERBO_REGULAR (is-a VERBO)
  (role concrete)
  (pattern-match reactive)
  (slot palabra (create-accessor write))
  (slot patron (create-accessor write)))
```

La clase VERBO_REGULAR hereda de la clase VERBO, el atributo palabra, este atributo almacena todas las palabras que pueden identificarse como verbos en infinitivo, debido a que los verbos regulares no presentan una variación significativa en su forma de ser escritos, el atributo patrón almacena las primeras letras que pueden identificar a un verbo conjugado en cualquier tiempo y cualquier persona.

4.1.8 VERBO_IRREGULAR

Otro tipo de verbos son los verbos irregulares, los cuales sufren variaciones visibles en su forma de ser escritos al ser conjugados. La estructura que define a un verbo irregular, es la siguiente:

```
(defclass VERBO_IRREGULAR (is-a VERBO)
  (role concrete)
  (pattern-match reactive)
  (slot palabra (create-accessor write))
  (multislot presente (create-accessor write))
  (multislot pasado (create-accessor write))
  (multislot futuro (create-accessor write)))
```

La clase VERBO_IRREGULAR hereda de la clase VERBO el atributo palabra, este atributo almacena todas las palabras que pueden identificarse como verbos en infinitivo, debido a que los verbos irregulares presentan una variación significativa en su forma de ser escritos, los atributos presente, pasado y futuro almacenan los tres tiempos principales del verbo para su posterior análisis.

4.1.9 Conjunción

Las conjunciones al igual que las preposiciones tienen una función especial en la interrelación de palabras e ideas, asociándolas y dándoles un sentido más amplio en su ordenamiento lógico. La estructura que define a una conjunción, es la siguiente:

```
(defclass CONJUNCION (is-a PALABRA)
  (pattern-match reactive)
  (role concrete)
  (multislot palabra (create-accessor write)))
```

4.1.10 Adverbio

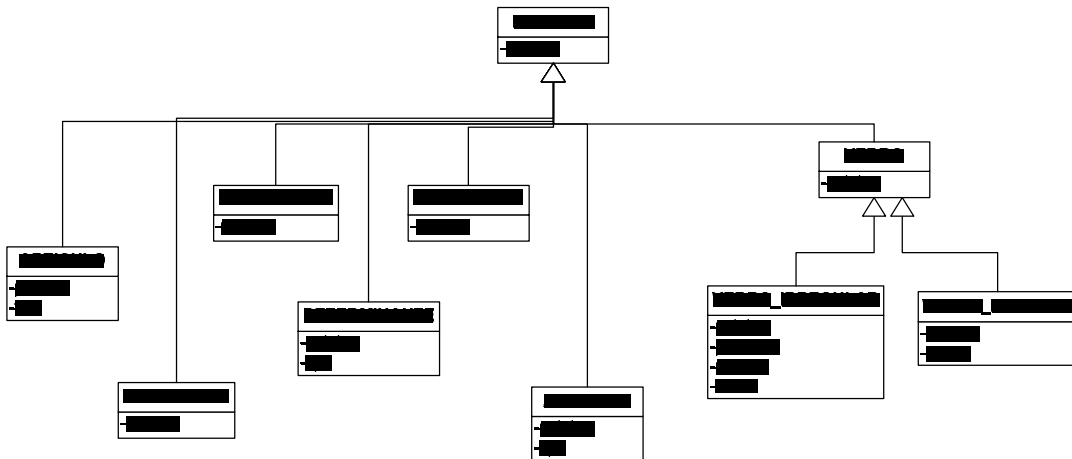
Los adverbios son palabras que determinan y califican a los verbos, existen diversos tipos de adverbios. La estructura que define a un adverbio, es la siguiente:

```
(defclass ADVERBIO (is-a PALABRA)
  (pattern-match reactive)
  (role concrete)
  (slot palabra (create-accessor write))
  (slot tipo (create-accessor write)))
```

La clase adverbio hereda de la clase PALABRA, el atributo palabra, este atributo almacena todas las palabras que pueden identificarse como adverbios, además en el atributo tipo es almacenado el tipo de adverbio que se describe en el atributo palabra.

Todos los objetos anteriormente descritos por medio de su estructura en pseudo-código, no describen la relación que mantienen entre ellos, ya que algunos de estos objetos son dependientes de otros, por lo cual en la figura 5 muestran los objetos anteriormente definidos y sus relaciones por medio de un diagrama de clases.

Figura 5. Diagrama de clases de palabras



Antes de terminar con esta sección debe de tomarse en cuenta la notación utilizada para la figura que se muestra en la parte superior, a fin de

comprender con un amplio margen de satisfacción. Como se mencionó anteriormente, la figura muestra un diagrama de las clases y su estructura, pero también muestra la relación existente entre ellas. La clase PALABRA es la raíz de todas las otras clases, esto se puede deducir fácilmente en el diagrama por medio de la flecha con punta vacía que indica la relación entre una clase hija y una clase padre, ya que la flecha vacía apunta hacia la clase padre desde la clase hija.

4.2 Hechos y variables globales

Además del lexicón como elemento principal de la base de hechos, el SRLN utiliza elementos en su base de hechos que permiten un correcto funcionamiento del mismo, en este apartado se exploran los hechos que forman parte de la base de hechos del SRLN y las variables globales que se utilizaron para controlar algunas tareas relacionadas en el SRLN.

4.2.1 Hechos

Algunos de los hechos utilizados en el SRLN y que no forman parte del lexicón del mismo, son explorados a continuación.

4.2.1.1 Definición de Hechos

Los hechos creados en *defacts hechos* son *linea* y *linea_aux*, estos hechos son útiles para la lectura del archivo de entrada que contiene la propuesta de ley planteada por el usuario legislador, almacenándose en los mismos las palabras que serán analizadas y estructuradas por el SRLN.

(defacts hechos

(linea)

(linea_aux))

4.2.1.2 Definición de Plantillas

Las plantillas definidas en el SRLN, se crearon a fin de obtener información en forma de hechos que sirviera de apoyo al reconocimiento de las palabras y su ordenamiento para la oración de salida.

La plantilla WORD se define con dos atributos que son línea y orden, el atributo línea almacena en cada instancia de la plantilla una palabra de cada una de las que son ingresadas por el usuario experto. A continuación se muestra la definición de la plantilla de WORD:

```
(deftemplate WORD
(slot línea)
(slot orden))
```

Además de la plantilla que almacena cada una de las palabras de la propuesta de los legisladores (WORD), se utilizó una plantilla más que ayuda a equiparar en las reglas el orden en el que se han colocado las palabras del legislador. Esta es la definición de la plantilla que ayuda a controlar el orden de las palabras:

```
(deftemplate ORDEN
(slot ordenar))
```

Como se indicó al inicio de este trabajo, las oraciones se dividen en dos partes que son el sujeto y el predicado, de manera casi general, es por ello necesario que el SRLN identifique este orden, a fin de proporcionar las salidas al SE de manera eficiente. Además del orden en el que se han colocado las palabras de la oración, se utiliza un control especial de la parte de la oración que se analiza para su salida.

El hecho que ayuda a controlar la parte de la oración que esta en análisis se define por medio de la clase ORDEN_ORACION, en la cual su único *atributo parte_oracion* indica en que parte se encuentra el análisis de la oración, si se encuentra sujeto o predicado, los valores que puede tomar este atributos son: 1 = sujeto, 2 = predicado. A continuación se muestra la definición de la clase ORDEN_ORACION:

```
(defclass ORDEN_ORACION (is-a USER)
  (pattern-match reactive)
  (role concrete)
  (slot parte_oracion (create-accessor write)(pattern-match reactive)))
```

Como parte del control en el reconocimiento de los patrones y palabras en el SRLN se ha definido una clase que indica el fin de la lectura del archivo que almacena la propuesta de ley del experto legislador. A continuación se muestra la estructura de esta clase:

```
(defclass FIN_ARCHIVO (is-a USER)
  (pattern-match reactive)
  (role concrete)
  (slot fin_archivo (create-accessor write)(pattern-match reactive)))
```

Por último, pero no por eso menos importante, se verá la clase que define la oración de salida que es igualmente utilizada por el SE. Esta clase no es exclusiva del SRLN, pero es la salida esperada que ha de generar el SRLN. Como se puede observar en su definición, esta almacena elementos importantes de la oración, sin embargo no son utilizados todos los tipos de palabras existentes, sino solamente aquellas que son de uso específico en el SE.

Luego de la equiparación de las reglas respectivas, que permiten el ordenamiento lógico de las palabras para su funcionamiento en el SE, se obtiene una instancia de la clase ORACION, la cual define en palabras del SE,

las ideas transmitidas por el legislador en lenguaje natural. A continuación se muestra la definición de la clase ORACION:

```
(defclass ORACION (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (multislot sustantivo_sujeto (create-accessor write))
  (multislot verbo (create-accessor write))
  (slot condicional (create-accessor write))
  (multislot sustantivo_predicado (create-accessor write))
  (slot neg_sustantivo_sujeto (create-accessor write)))
```

- **verbo:** En este atributo son almacenados los verbos identificados en la propuesta de ley ingresada por el legislador, por ejemplo en la oración: “Toda persona es libre en dignidad y derechos”, se obtendrá como resultado el almacenamiento del verbo *ser* en el atributo verbo.
- **condicional:** Este atributo almacena la condición encontrada en el predicado, en el caso de existir tal condición, de lo contrario almacenara un valor nulo. Si existiera una condición, como el siguiente ejemplo: Ninguna persona puede ser detenida o presa, sino por causa de delito o falta, dará como resultado el almacenamiento de la palabra “sino” en el atributo condicional, debe de remarcar aquí que por condicional se referiría en la oración del legislador a la palabra identificada como un adverbio de tipo condicional.
- **sustantivo_predicado:** En este atributo son almacenados los sustantivos del predicado, por ejemplo: Toda persona es libre en dignidad y derechos, lo cual dará como resultado el almacenamiento de las palabras “dignidad” y “derechos” en el atributo sustantivo_predicado.
- **neg_sustantivo_sujeto:** Este atributo contendrá la negación encontrada en el sustantivo del sujeto, en el caso de existir tal negación, de lo contrario almacenara un valor nulo. Si existiera una negación, como el

siguiente ejemplo: No toda persona es libre en dignidad y derechos, dará como resultado el almacenamiento de la palabra “no” en el atributo `neg_sustantivo_sujeto`. Nótese que lo que se ha dado por llamar negación del sustantivo del sujeto, será en realidad en el idioma español un adverbio de tipo negación.

4.3 Reglas

El proceso de análisis de las entradas brindadas por el usuario experto no sería posible en el SRLN, sin haberse definido reglas que definan la lógica de dicho análisis, a continuación se muestra la definición de cada regla y una breve descripción de las mismas.

4.3.1 Inicio

```
(defrule inicio =>
(load-instances instancias.txt)
(open "C:\\propuesta.txt" archivo "r")
(bind ?linea (readline archivo))
(printout t "Leyendo archivo..." crlf)
(while (not(eq EOF ?linea))
(bind ?*ordenar* (+ ?*ordenar* 1))
(assert (WORD (linea (lowercase ?linea))(orden ?*ordenar*)))
(bind ?linea (readline archivo)))
(close archivo)
(assert (ORDEN (ordenar 1)))
(make-instance FIN of FIN_ARCHIVO (fin_archivo 1))
(printout t "Se leyó el archivo" crlf))
```

Esta es la primer regla que debe ser activada por el SRLN, ya que es en esta regla que se hace la lectura por medio de un archivo, de la frase ingresada

por el usuario. En cada nueva línea del archivo se encuentra almacenada una palabra la cual es almacenada en la base de hechos y ordenada conforme se van leyendo las palabras. Al finalizar la lectura del archivo es creado un hecho que indica la finalización de la lectura del mismo.

4.3.2reconoce_articulo

```
(defrule reconoce_articulo
(declare (salience 30))
?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
?b <- (object (is-a ARTICULO)(palabra ?x))
?c <- (WORD (línea ?y)(orden ?z))
?d <- (ORDEN (ordenar ?o))
(test (eq ?z ?o))
(test(= (str-compare ?x ?y) 0))
=>
(modify ?d (ordenar (+ ?o 1)))
(printout t "Encontro articulo " ?y " , " ?x crlf)
(retract ?c))
```

En esta regla se busca reconocer algún patrón que se encuentre identificado como un artículo dentro del lexicón, luego de haberse encontrado el fin del archivo de entrada. La regla no almacena el artículo reconocido, este únicamente se halla dentro del listado de lexemas válidos y la regla permite que el proceso pase a la siguiente palabra que debe ser analizada.

4.3.3reconoce_preposicion

```
(defrule reconoce_preposicion
?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
?b <- (object (is-a PREPOSICION)(palabra ?x))
?c <- (WORD (línea ?y)(orden ?z))
```

```
?d <- (ORDEN (ordenar ?o))
(test (eq ?z ?o))
(test(= (str-compare ?y ?x) 0))
=>
(modify ?d (ordenar (+ ?o 1)))
(printout t "Encontro la preposicion " ?y ", " ?x crlf)
(retract ?c)
```

En esta regla se busca reconocer algún patrón que se encuentre identificado como una preposición dentro del lexicón, luego de haberse encontrado el fin del archivo de entrada. La regla no almacena la preposición reconocida, esta únicamente se halla dentro del listado de lexemas válidos y la regla permite que el proceso pase a la siguiente palabra que debe ser analizada.

4.3.4reconoce_verbo_irregular_presente

```
(defrule reconoce_verbo_irregular_presente
?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
?b <- (object (is-a VERBO_IRREGULAR)(presente ?x))
?c <- (WORD (linea ?y)(orden ?z))
?d <- (ORDEN (ordenar ?o))
(test (eq ?z ?o))
(test(neq (str-index ?y ?x) FALSE))
=>
(bind ?*orden_oracion* 1)
(printout t "Encontro verbo irregular presente " ?y ", " ?x crlf)
(modify ?d (ordenar (+ ?o 1)))
(retract ?c)
```

Esta regla identifica algún verbo irregular que se encuentre dentro del lexicón del SRLN, específicamente verifica si alguno de los verbos irregulares que ha ingresado el usuario se encuentra en tiempo presente, de ser así,

almacena la forma del verbo en infinitivo, es decir, si el verbo en presente dice “estoy” el SRLN almacenará “estar”.

4.3.5reconoce_verbo_irregular_pasado

```
(defrule reconoce_verbo_irregular_pasado
?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
?b <- (object (is-a VERBO_IRREGULAR)(pasado ?x))
?c <- (WORD (linea ?y)(orden ?z))
?d <- (ORDEN (ordenar ?o))
(test (eq ?z ?o))
(test(neq (str-index ?y ?x) FALSE))
=>
(bind ?*orden_oracion* 1)
(printout t "Encontro verbo irregular presente " ?y " , " ?x crlf)
(modify ?d (ordenar (+ ?o 1)))
(retract ?c))
```

Esta regla identifica algún verbo irregular que se encuentre dentro del lexicón del SRLN, específicamente verifica si alguno de los verbos irregulares que ha ingresado el usuario se encuentra en tiempo pasado, de ser así, almacena la forma del verbo en infinitivo, es decir, si el verbo en pasado dice “estaba” el SRLN almacenará “estar”.

4.3.6reconoce_verbo_regular

```
(defrule reconoce_verbo_regular
?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
?b <- (object (is-a VERBO_REGULAR)(palabra ?x)(patron ?p))
?c <- (WORD (linea ?y)(orden ?z))
?d <- (ORDEN (ordenar ?o))
(test (eq ?z ?o))
```

```
(test(neq (str-index ?p ?y) FALSE))
=>
(bind ?*orden_oracion* 1)
(printout t "Encontro verbo regular " ?y " , " ?x crlf)
(modify ?d (ordenar (+ ?o 1)))
(slot-insert$ FRASE verbo 30 ?x)
(retract ?c)
```

Esta regla identifica algún verbo regular que se encuentre dentro del lexicón del SRLN, específicamente verifica el verbo ingresado por medio de reconocimiento de patrones, teniendo un patrón básico para cada verbo regular, se busca dicho patrón dentro de ingresada por el usuario dará, este emparejamiento de patrones dará por resultado el verbo regular esperado en infinitivo.

4.3.7reconoce_sustantivo

```
(defrule reconoce_sustantivo
(declare (salience 10))
?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
?b <- (object (is-a SUSTANTIVO)(palabra ?x))
?c <- (WORD (linea ?y)(orden ?z))
?d <- (ORDEN (ordenar ?o))
(test (eq ?z ?o))
(test(= (str-compare ?x ?y) 0))
=>
(printout t "Encontro sustantivo " ?y " , " ?x crlf)
(modify ?d (ordenar (+ ?o 1)))
(if (eq ?*orden_oracion* 0) then
(slot-insert$ FRASE sustantivo_sujeto 30 ?y)
else
(slot-insert$ FRASE sustantivo_predicado 30 ?y))
(retract ?c)
```


Regla que identifica algún sustantivo registrado dentro del lexicón del SRLN, esta regla es importante para organizar la frase que se obtiene como salida del SRLN. Específicamente esta regla reconoce algún sustantivo dentro de la frase ingresada por el usuario experto, al encontrar un sustantivo dentro de la frase de entrada, se inserta en el atributo sustantivo_sujeto de la clase frase el nuevo sustantivo que se ha encontrado siendo esa la salida de la regla.

4.3.8reconoce_adverbio

```
(defrule reconoce_adverbio
(declare (saliency 10))
?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
?b <- (object (is-a ADVERBIO)(palabra ?x)(tipo negacion))
?c <- (WORD (linea ?y)(orden ?z))
?d <- (ORDEN (ordenar ?o))
(test (eq ?z ?o))
(test(= (str-compare ?x ?y) 0))
=>
(printout t "Encontro adverbio " ?y ", " ?x crlf)
(modify ?d (ordenar (+ ?o 1)))
(if (eq ?*orden_oracion* 0) then
(send [FRASE] put-neg_sustantivo_sujeto ?y)
else
(send [FRASE] put-neg_sustantivo_predicado ?y))
(retract ?c))
```

Regla que identifica algún adverbio registrado dentro del lexicón del SRLN, esta regla es importante para organizar la frase que se obtiene como salida del SRLN. Específicamente esta regla reconoce algún adverbio que sea de tipo negación dentro de la frase ingresada por el usuario experto.

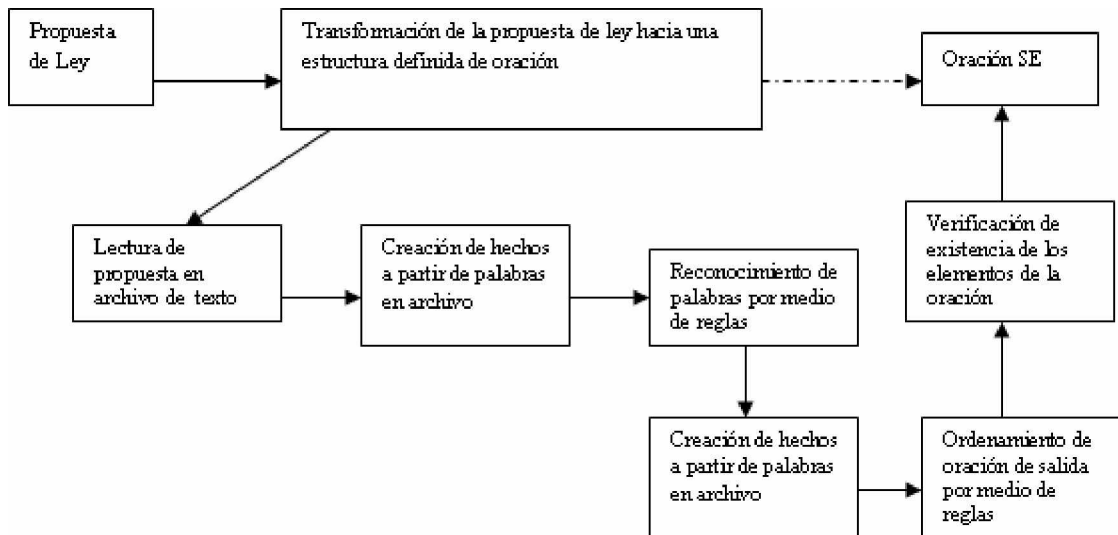
Al encontrar un adverbio de negación dentro de la frase de entrada, se inserta en el atributo neg_sustantivo_sujeto si forma parte del sustantivo de la

oración de entrada, o en el atributo `neg_sustantivo_predicado` si este forma parte del predicado de la oración de entrada. De no existir un adverbio de negación en la oración de entrada, estos atributos quedan con valores nulos en la FRASE de salida.

4.4 Proceso de equiparación de reglas

En este apartado del capítulo se verá el proceso de equiparación y el funcionamiento lógico que realiza el SRLN a fin de traducir el lenguaje del legislador (idioma español) hacia el lenguaje esperado del SE, es necesario tomar en cuenta que en el capítulo 1 se definió formalmente como Lengua Origen (LO) y Lengua Meta (LM) respectivamente.

Figura 6. Diagrama de estados del proceso de equiparación de reglas



El proceso de equiparación de las reglas en el sistema es en sí sumamente sencilla, ya que por cada palabra tipo de palabra existente en el idioma español (artículo, sustantivo, ...) es utilizada una regla, cada regla es

activada al encontrarse alguna palabra que cumpla las condiciones establecidas en el SRLN.

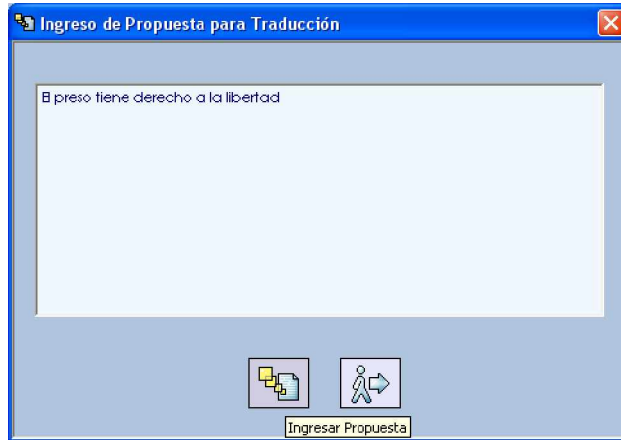
El proceso de equiparación de reglas comienza con la lectura de la regla de inicio, en esta regla es leído el archivo, ordenadas las palabras e inicializados los valores para el reconocimiento y posterior traducción de las palabras. El siguiente paso consiste en el reconocimiento de las palabras en cada regla, una vez encontrada una palabra por alguna de las reglas, en el orden correspondiente a cada palabra, en la parte derecha de las reglas se actualizan los valores del orden en el cual son analizadas las palabras y se almacena la palabra, según corresponda, en la estructura definida para la oración de salida que será utilizada por el Sistema Experto Legislativo.

La mejor manera de demostrar el proceso de activación de cada regla que analiza el contenido de la oración será por medio de un ejemplo: Supóngase que una de las palabras ingresadas en la propuesta de ley del usuario es *será*, como se puede observar esta palabra es un verbo que ha sido conjugado ya, lo cual da como resultado que el sistema trate de encontrar una palabra o un patrón (en el caso de los verbos) que cumpla con una de las condiciones del lado izquierdo de las reglas.

A fin de mostrar la funcionalidad del sistema, a continuación se muestra un ejemplo que le ilustrará de mejor manera:

- Suponer que se ha ingresado en la interfaz con el usuario la siguiente frase: *“El preso no tiene derecho a la libertad”*.

Figura 7. Pantalla de ingreso de Propuestas



- Luego del ingreso de la propuesta en la interfaz, se generará un archivo, el cual contiene las palabras encontradas en la frase
- Comienza el proceso de reconocimiento del SRLN, con la lectura del archivo, en este punto debe tomarse en cuenta que las instancias de las palabras utilizadas ya han sido definidas previamente para su reconocimiento en el SRLN.
- Al ser leído el archivo es activada la regla inicial *inicio*, para tal efecto, la información que se lee se almacena en la plantilla WORD, descrita en la sección 4.2.1.2, almacenando así la palabra y el orden que tiene en la frase, por ejemplo para la palabra *preso*, del ejemplo, se almacenaría en la plantilla con el siguiente formato (WORD (palabra preso)(orden 2)).
- Al finalizarse la lectura del archivo y ser almacenadas las palabras de las frases del usuario, comenzará el proceso de equiparación de las reglas que permiten el análisis y posterior ordenamiento de la oración de salida.
- Para este ejemplo la primer regla a activarse es la regla *reconoce_articulo*, la cual reconoce un articulo en la oración, y al ser un artículo la primer palabra de la oración de entrada, se activaría esta regla, es de hacer notar que para esta regla no se obtiene una salida en la estructura ORACION ya que en la misma no se contempla la

existencia de artículos, pero nótese que en el proceso de reconocimiento si es analizado, dando así lugar a su utilización en otros sistemas que si requieran de algunas palabras que no se incluyen en la oración de salida para este sistema.

- Continuando con el ejemplo se reconocerá la palabra *preso* como un sustantivo, al ser la segunda palabra, activándose así la regla *reconoce_sustantivo*, a diferencia del artículo que reconoce inicialmente con esta regla si se obtiene una salida en la oración.
- Y así sucesivamente se continuará con el proceso de equiparación hasta obtener una oración con la siguiente estructura:

[FRASE] of ORACION (sustantivo_sujeto "preso") (verbo "tener") (condicional nil) (sustantivo_predicado "derecho" "libertad") (neg_sustantivo_sujeto "no")

4.5 Aprendizaje del SRLN

El SRLN, cuenta con un conjunto básico de palabras en su lexicón, dichas palabras se encuentran clasificadas según el tipo de cada una, tal y como se definió anteriormente. Sin embargo, el prototipo que se enuncia en este trabajo de graduación, proporciona la probabilidad de agregar nuevas palabras que brinden un aprendizaje apropiado al sistema.

Este aprendizaje se hace posible a través de un módulo de interacción con el usuario, en el que el usuario puede ingresar la palabra que desea agregar, además el SRLN cuenta con un módulo de consulta para verificar las palabras de los diferentes tipos que han sido ingresados en la base de hechos, estos y otros beneficios que permiten el aprendizaje del SRLN son presentados en la siguiente sección.

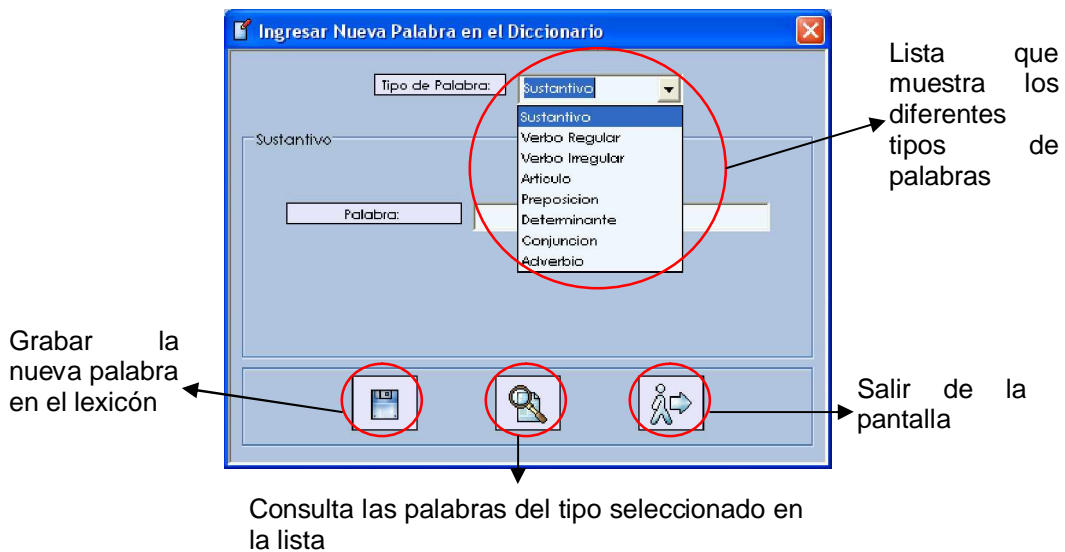
Debe de hacerse mención, que la información que es ingresada al sistema es almacenada en tablas de Access®, en donde cada una de las

palabras que son representadas en las tablas, son todas aquellas palabras que fueron definidas en la sección 4.1, es de hacer notar que por medio del uso de tablas en una base de datos, se implementa el medio de almacenamiento de las palabras, la lógica de la estructura del lexicón, puede observarse en la figura 5 de la sección 4.1.

4.5.1 Módulo de ingreso y consulta de nuevas palabras

Una de los alcances que brinda el SRLN que se describe en este trabajo de graduación, es la posibilidad de añadir nuevo conocimiento, de tal manera que el lexicón pueda hacerse robusto para el reconocimiento de las frases ingresadas por el usuario.

Figura 8. Ingreso y Consulta de nuevas palabras



En la figura 8 se muestra la pantalla del SRLN en la que cual se permite realizar un ingreso de una nueva palabra, o consultar todas las palabras según su clasificación en el lexicón. Como se puede apreciar en la figura, en la parte inferior de la misma, el usuario puede realizar un ingreso de una nueva palabra,

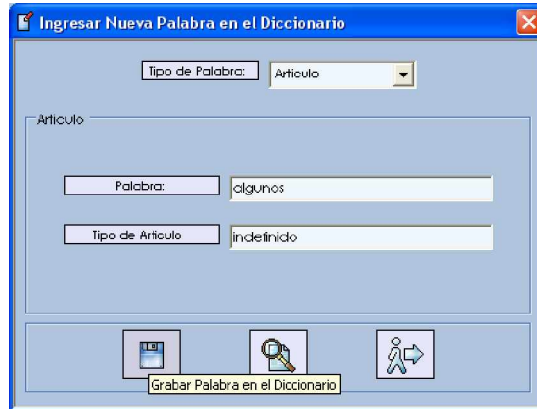
consultar todas las palabras según su clasificación en el lexicón o salir del módulo de Ingreso/Consulta de palabras. A fin de mostrar el proceso de aprendizaje y retroalimentación que brinda el SRLN, se expondrá a continuación la funcionalidad del módulo.

4.5.1.1 Ingreso de nuevas palabras en el lexicón

Debido a la existencia de diferentes tipos de palabras en el idioma español, fue necesario realizar una clasificación de las mismas, debido a esto no todas las palabras brindan la misma información en el SRLN, por ejemplo los artículos los hay de diferentes tipos (directos, indirectos), al igual que los adverbios, sin embargo los sustantivos no necesitan ser clasificados (al menos no para el SRLN), los verbos irregulares cuentan con diferentes conjugaciones y en los verbos regulares se puede observar un patrón indistintamente del tiempo y la persona en la que se realice una conjugación.

Así por ejemplo, en la figura 8, se muestra la pantalla de ingreso de algún sustantivo, como se puede observar, en esta pantalla únicamente se cuenta con una caja de texto en la que se debe ingresar alguna palabra que sea un sustantivo, sin embargo, no todos los tipos de palabras necesitan almacenar la misma información, por ejemplo, en la figura 9 se muestra el ingreso de un artículo, para este tipo de palabra es necesario almacenar la palabra y el tipo de artículo que es, añadiéndose en el formulario un campo adicional (el tipo) que necesita ser almacenado en el lexicón.

Figura 9. Ingreso de un artículo al lexicón



The screenshot shows a window titled "Ingresar Nueva Palabra en el Diccionario". At the top, there is a dropdown menu labeled "Tipo de Palabra:" with "Artículo" selected. Below this, there is a section labeled "Artículo" containing two input fields: "Palabra:" with the text "algunos" and "Tipo de Artículo:" with "indefinido" selected. At the bottom of the window, there are three icons: a document, a magnifying glass, and a person with an arrow. Below these icons is a button labeled "Grabar Palabra en el Diccionario".

4.5.1.2 Consulta de palabras del lexicón

En la sección anterior se mostró el ingreso de las palabras en el SRLN, pero, en ocasiones, es necesario consultar las palabras que ya forman parte del lexicón y aquellas palabras que son ingresadas por el usuario experto. De esta manera se puede consultar el nivel de conocimiento adquirido por el SRLN, es decir, se puede solicitar al sistema, que informe acerca del conocimiento que ha adquirido, a través de la base de datos de su lexicón.

En la figura 10 se muestra una pantalla, en la que los resultados obtenidos, son del tipo verbo irregular, en la pantalla de ingreso y consulta principal (ver figura 8), el usuario, únicamente debe seleccionar el tipo de palabra del que desea obtener información, y posteriormente, hacer clic en el botón de consulta (botón central del formulario), mostrando así, los resultados contenidos en el lexicón, sobre el tipo de palabra verbo irregular (figura 10).

Figura 10. Pantalla de resultado para los verbos irregulares

Verbo Irregular			
PALABRA	PRESENTE	PASADO	FUTURO
tener	tiene	tenía	tendrá
ser	es	fue	será
estar	esta	estuvo	estará

Sin embargo, esta información probablemente sea la información original que almacena el SRLN en su lexicón, ¿de que manera crecerá entonces el conocimiento del SRLN?

Por medio del proceso de grabación, descrito en la sección anterior, es la forma en la que es proporcionada, por parte del usuario, nueva información para el lexicón del SRLN, supóngase, que una persona ingresó un nuevo artículo al lexicón (figura 9), y la siguiente vez, que el usuario se encuentre en la pantalla de ingreso y consulta, desee consultar que efectivamente se ha grabado el artículo que ingreso, entonces debe seleccionar el tipo de palabra (artículo, en este caso), y finalmente debe hacer clic en el botón de consulta, mostrándose así, una pantalla con los valores posibles, para el tipo de palabra escogido por el usuario (figura 11). Como se puede observar en la figura 11, aparece marcada la palabra que fue ingresada por el usuario, dicha palabra es un artículo, más explícitamente esa palabra es “algunos” y es de tipo indefinido.

Figura 11. Consulta de un artículo ingresado por el usuario



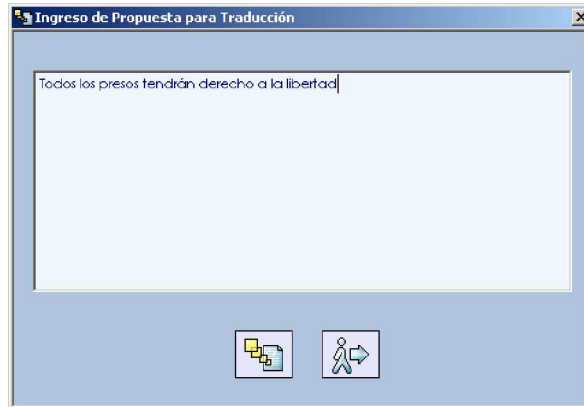
4.6 Ejemplos de Ejecución

Con el propósito, que el lector, comprenda de una mejor manera el funcionamiento del SRL, a continuación, son ilustrados dos ejemplos de la forma en la que son ingresadas las frases en el Sistema y las salidas que éste proporciona, es decir, que resultado se obtiene en el archivo de salida, que servirá para su posterior interpretación por un Sistema Experto. A fin de obtener dos ilustraciones de la forma en la cual funciona el SRLN, se mostrará un ejemplo de una propuesta con una premisa afirmativa y otro ejemplo de una premisa de negación.

4.6.1 Ejemplo de Oración de Entrada Afirmativa

En la pantalla de ingreso de la propuesta el usuario ingresa la siguiente frase: Todos los presos tendrán derecho a la libertad.

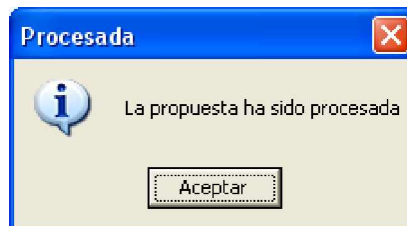
Figura 12. Pantalla de ingreso de propuestas (Ejemplo 1)



Luego de ingresarse en la pantalla del SRLN la propuesta, por parte del usuario, es analizada la misma, mientras la propuesta es analizada la pantalla de ingreso será bloqueada, visualizándose una pantalla de ejecución de comandos.

Al finalizar el análisis de la frase ingresada por el usuario, en la pantalla de ingreso de propuestas, se cerrará automáticamente la pantalla de ejecución de comandos y el Sistema indicará la finalización del análisis de la propuesta ingresada.

Figura 13. Aviso de finalización de análisis



Tal y como indicará el sistema al finalizar su ejecución, las propuestas expresadas por medio de frases en idioma español, son traducidas a un lenguaje que puede ser reconocido por otro tipo de sistema, como lo es un

sistema experto. La traducción se realiza en un archivo de texto que contiene el Lenguaje Meta (LM). Para el ejemplo propuesto en esta sección el archivo de salida sería la siguiente:

```
([FRASE] of ORACION
(sustantivo_sujeto "presos")
(verbo_aux nil)
(verbo_modal nil)
(verbo tener)
(adverbio)
(condicional nil)
(sustantivo_predicado "derecho" "libertad")
(neg_verbo nil)
(neg_sustantivo_sujeto nil)
(neg_sustantivo_predicado nil))
```

Al hacer un análisis breve de la oración de salida, se puede observar que no son tomadas en cuenta todas las palabras de la oración que fue ingresada al sistema, sin embargo las que en este ejemplo fueron obtenidas son suficientes para el posterior análisis que se puede realizar en un sistema experto, como se indicó en capítulos anteriores, este es un sistema de traducción que puede ser útil para cualquier otro Sistema experto que base su análisis en un lenguaje meta definido, por lo que el que no se hayan incluido todas la palabras de la oración de origen en la oración destino (lenguaje meta), no quiere decir que el SRLN no las haya reconocido, al contrario, fueron exitosamente reconocidas pero no incluidas en la oración de salida de este prototipo.

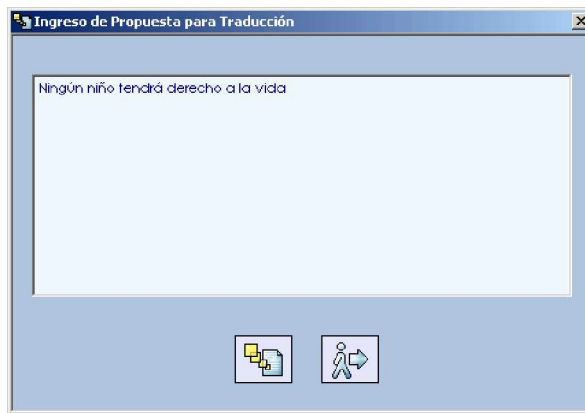
4.6.2 Ejemplo de Oración de Entrada con Negación

A fin de demostrar la funcionalidad del sistema y las variantes que puede observar con respecto al análisis que realiza durante la traducción de las oraciones de entrada, en este ejemplo se muestra una oración con una

negación, lo cual es importante para un sistema experto legislativo el cual necesita conocer las contradicciones que puede tener una propuesta de ley con respecto a una ley de la constitución de la República de Guatemala.

Supóngase que el usuario ingresa la siguiente frase al sistema: Ningún niño tendrá derecho a la vida (ver figura 12).

Figura 14. Pantalla de ingreso de propuestas (Ejemplo 2)



Luego del procedimiento de análisis, descrito en el apartado anterior, el sistema ofrecerá la siguiente salida, en el archivo correspondiente para el posterior análisis del significado de la frase en otro Sistema Experto:

```
([FRASE] of ORACION
(sustantivo_sujeto "niño")
(verbo_aux nil)
(verbo_modal nil)
(verbo tener)
(adverbio)
(condicional nil)
(sustantivo_predicado "derecho" "vida")
(neg_verbo nil)
(neg_sustantivo_sujeto "ningun")
(neg_sustantivo_predicado nil))
```

Como se puede observar en la salida ofrecida por el sistema existe únicamente una diferencia con respecto al ejemplo anterior, la cual es la negación de la oración, debido a que la oración del usuario contenía el adverbio de negación *Ningún*, como se puede observar en la salida de este ejemplo, esta negación es almacenada en el atributo `neg_sustantivo_sujeto`, lo cual permite identificar, en el sistema experto que realice el análisis de las salidas ofrecidas por este SRLN, el significado que se deseaba poner en la propuesta ingresada por el usuario.

5 FUNDAMENTOS DE USO, FUNCIONAMIENTO Y APORTES

Como es de esperarse con este trabajo, se concluyó con la obtención de un prototipo de Sistema Experto para Reconocimiento de Lenguaje Natural y como tal existen características y elementos que pueden ser adicionados al mismo, con este prototipo se han sentado las bases para la construcción de un sistema más robusto, sin embargo, este es un prototipo funcional con lo cual se obtienen, no todas, pero las características esenciales esperadas para un sistema final.

En este capítulo se muestran los requerimientos que fueron necesarios para la construcción del prototipo, los límites y alcances, el impacto del mismo y los aportes que ha brindado, para concluir con una exploración de las sugerencias para posteriores mejoras y ampliaciones que pueden realizarse sobre el prototipo.

5.1 Requerimientos

Los requerimientos con los que un sistema debe cumplir son esenciales, ya que proveen un análisis del punto al que se quiere llegar, para el caso de este prototipo ya fueron listados algunos requerimientos del sistema, es decir, la forma que se deseaba que tomara el prototipo de este trabajo de graduación, pero también es importante destacar requerimientos que son importantes para el desarrollo, que van desde requerimientos del hardware que se utilizarían hasta requerimientos de recursos humanos, para identificar a los interesados que participarán de una u otra manera con el prototipo y así realizar un perfil de los mismos.

5.1.1 Requerimientos de Hardware

Para la construcción del prototipo del SRLN es necesaria una computadora con requerimientos mínimos de procesador de 600 MHz, 64 Mb de memoria RAM y disco duro de 20 Gb. Dependiendo del crecimiento de la base de hechos, la cantidad de espacio en el disco duro puede aumentar con el tiempo, dependiendo del tamaño y modificaciones realizadas a la base de hechos y el lexicón.

5.1.2 Requerimientos de Software

Los requerimientos de software para el prototipo de SRLN se pueden distinguir entre requerimientos con los que se debe cumplir para la ejecución y además para el desarrollo del SRLN.

Para el desarrollo exitoso del prototipo fue necesario contar con la herramienta EHSIS ver. 1.03, para el reconocimiento del lenguaje natural utilizando reglas, y Visual Basic .NET® para la interfaz gráfica con el usuario, además del sistema operativo Windows® 9X o mayor. La base de datos que almacena los valores de los diferentes tipos de palabras y da forma al lexicón del SRLN, fue implementada en Access 2003 ®

Para la ejecución del prototipo ya desarrollados es necesario contar con EHSIS 1.03 y el sistema operativo Windows® 9X o mayor.

5.1.3 Recursos Humanos

Es de suma importancia la participación de un experto en el área de la lingüística que apoye a la ampliación del prototipo que se presenta con este trabajo de graduación.

Para el desarrollo o mejoras que se deseen realizar para el SRLN es necesario que la persona encargada de estas tareas, tenga conocimientos de programación orientada a objetos, Sistemas Expertos, reconocimiento de lenguaje natural y un conocimiento básico y general de la constitución de la República de Guatemala por lo menos en sus primeros 10 artículos. Es importante remarcar en este punto, que tanto la persona que realice el desarrollo del SRLN o su ampliación, interactúe de una manera continua con el experto en lingüística a fin de enriquecer el aprendizaje de este prototipo y también la estructura de las palabras que interpreta y transforma, así como el proceso de reconocimiento que conlleva esta interpretación y transformación.

Para la ejecución del sistema es necesario que la persona que haga uso del mismo cuente con conocimiento básico del uso de la computadora y de programas en ambiente Windows®, además de ser un experto (aún en formación o profesional) del derecho, a fin de ingresar propuestas de ley estructuradas al sistema por medio de una interfaz gráfica en el sistema operativo mencionado.

5.2 Límites y Alcances

En esta sección se analiza el funcionamiento que puede ofrecer el sistema y los puntos que no han sido cubiertos para este prototipo, más adelante se describe como pueden ser cubiertos algunos de los límites con los que cuenta el SRLN.

5.2.1 Límites

Los límites son aquellos puntos que aún al ser cubiertos por el sistema tienen una restricción, para este prototipo SRLN muchos de estos límites pueden ser superados, para lo cual se plantean más adelante sugerencias para optimizar el prototipo en el futuro.

- La oración de salida esta compuesta únicamente por algunos elementos que conforman una oración en idioma español, no por todos, en esta medida que la oración fue adecuada para su interpretación por parte del Sistema Experto Legislativo al que complementará con esta entrega
- No se indican los tiempos ni la persona en los verbos, es decir, las conjugaciones son omitidas para la construcción de la oración de salida, obteniéndose así únicamente verbos en infinitivo para su posterior análisis por otros sistemas, dado que lo que se desea conocer es la acción, sin importar la conjugación o el tiempo del verbo.
- El prototipo no identifica aún las relaciones que puedan darse entre dos oraciones independientes del experto legislador, es decir si es analizada una oración inicialmente se obtendrá una oración de salida de la misma y a continuación se analiza una segunda oración que esta vinculada a la primera, este prototipo no podrá identificar la relación con exactitud.

5.2.2 Alcances

En esta sección del capítulo se analizan los puntos sobre los cuales el sistema se desempeña y en los cuales se ha logrado obtener una visión amplia hacia la consecución de los objetivos planteados inicialmente al desarrollo de este prototipo.

- Se ha logrado definir una estructura que pueda ser fácilmente adecuada o utilizada para la construcción de sistemas que requieran de la interpretación del idioma español para ser utilizado en procesos, como el análisis de leyes.
- Con este prototipo se ha creado un lexicón con los tipos de palabras más utilizadas comúnmente en el habla humana.
- La estructura de la base de hechos, la cual esta principalmente formada por el lexicón creado, cuenta con la característica que permite hacer crecer a la base de hechos, debido a las facilidades que los objetos proveen para tal efecto.
- El prototipo provee un medio de comunicación entre la persona y la máquina, de tal manera que ha permitido estructurar el idioma por medio de un formato específico, lo cual permite que una persona experta pueda comunicarse por medio de este prototipo, con un sistema especializado en la materia de estudio del experto humano.

5.3 Impacto esperado y Aportes del SRLN

Los sistemas de traducción automática tienen diversos usos en distintas aplicaciones de la vida, tales como el reconocimiento de voz y la traducción de textos de un idioma hacia el otro, de cualquier manera la aplicación de este tipo de sistemas ha sido restringida y difícilmente desarrollada, este prototipo en sí no pretende enfocarse a algunas de estas áreas de aplicación mencionadas, sino abrir nuevas oportunidades a su desarrollo en un área tan compleja para ser representada por medio de un sistema, como lo es la estructura lingüística utilizada en la legislación guatemalteca, para ser más conciso, en el lenguaje utilizado por legisladores expertos.

El impacto que se espera, entonces para este sistema, no se remite a la construcción de un SRLN que sirva de interfaz entre el usuario y un Sistema Experto Legislativo, sino que se centra en la construcción de un sistema de traducción entre el idioma español y cualquier sistema que pueda ser adecuado a una gramática (una oración en el caso de las salidas de este prototipo), de tal manera que al existir un sistema que necesite comunicarse con un experto en alguna materia no necesite pasar por el proceso del reconocimiento de las frases del usuario, sino que utilice la herramienta que con este trabajo de graduación se propone.

Con este trabajo de graduación se ofrece una adición de conocimiento que enriquezca el estudio de la inteligencia artificial, la lingüística y en un menor grado el derecho constitucional en Guatemala.

CONCLUSIONES

1. A pesar de la extensión del habla humana y la complejidad que esta puede mostrar, constituye un sistema, si bien complejo, no imposible de representar por medio de un sistema cerrado, como lo es un programa de computadora, muestra de ello son las diferentes aplicaciones para procesamiento de lenguaje natural que puede encontrarse, fácilmente, especialmente, a través de Internet.
2. La traducción automática, de acuerdo a un buen número de documentos, es una metodología de procesamiento de textos que cuenta con un buen número de deficiencias, aunque, también, se reconoce que es una herramienta importante para el desarrollo de la Inteligencia Artificial y es por medio de esta metodología que se ha logrado en este trabajo obtener los resultados esperados, al traducir porciones del lenguaje humano, hacia un conjunto de patrones, fácilmente, reconocidos por un Sistema Experto Legislativo.
3. La base de hechos, sobre la cual se fundamenta un sistema de traducción automática, provee un importante mecanismo de reconocimiento de patrones en una lengua origen, para su posterior traducción hacia una lengua meta y, a la vez, la representación de conocimiento en estas bases de hechos es un medio sencillo para tal efecto.

4. Una metodología básica en la Inteligencia Artificial para la representación del conocimiento la constituye el uso de los marcos, es a través de esta metodología que se ha logrado definir un lenguaje meta que describa las oraciones de un humano, usuario del sistema, para la alimentación de un Sistema Experto Legal.

5. Debido a la complejidad que el habla humana representa, a pesar de que la orientación de este trabajo se pretende hacia el reconocimiento de propuestas de ley, se ha descubierto que no, únicamente, en esta área de estudio se puede enmarcar el habla humana, sino que cualquier palabra que pueda expresar un ser humana debe ser tomada en consideración para un análisis completo y exitoso de cualquier texto para ser traducido.

RECOMENDACIONES

1. Debido a que el sistema que se muestra a través de este trabajo de graduación es un sistema experto, como tal, tiene la capacidad inherente de aprender y razonar por sí mismo, por lo tanto el usuario del sistema debe de ser un experto en el idioma español de tal manera que pueda guiar al sistema a través de su aprendizaje en el tiempo.
2. La inteligencia artificial es un campo de estudio lleno de contradicciones, tales como la corriente que cree que la inteligencia artificial puede desplazar al ser humano, es decir, la inteligencia artificial fuerte; y aquella corriente que asegura que las máquinas nunca podrán llegar al nivel del ser humano, es decir, la inteligencia artificial débil. Cualquiera que sea la corriente de pensamiento, es notable, que de una u otra manera, el ser humano debe de verse beneficiado con sistemas inteligentes, como el descrito en este trabajo de graduación, sistemas que faciliten tareas, como lo es la traducción automática. Es por ello, recomendable, ampliar el campo de la inteligencia artificial, tanto en el reconocimiento de lenguaje natural, como en otras disciplinas, como lo es el derecho constitucional, entre otros.
3. En este trabajo de graduación, el sistema que se programó, se ejecuta en diferentes plataformas de desarrollo, como lo son EHSIS, Visual Basic.NET y Access. Es importante para la continuación del desarrollo de el prototipo descrito en este trabajo, que se continúe con una buena administración del proyecto, es decir, que para el crecimiento y mayor

alcance del SRLN, el desarrollador puede hacer modificaciones a su funcionamiento. Sin embargo, es necesario que para las modificaciones o mejoras al SRLN, se cuente con el conocimiento debido de las herramientas de desarrollo, que fueron descritas previamente y fueron utilizadas para el SRLN.

4. Siendo el reconocimiento del lenguaje natural, una rama de la inteligencia artificial y, ya que, en la Escuela de Ingeniería en Ciencias y Sistemas, se contempla dentro del pensum de estudios la inteligencia artificial, es recomendable que se enseñe el reconocimiento del lenguaje natural y las técnicas que este utiliza para llevarse a cabo y las aplicaciones que cada técnica tiene en la vida real, tal y como lo es la traducción automática, tratada en este trabajo de graduación, en la que es utilizada como una metodología de traducción entre el lenguaje humano y un lenguaje intermedio para un sistema computacional.

BIBLIOGRAFÍA ELECTRÓNICA

1. <http://www.scism.sbu.ac.uk/inmandw/tutorials/nlp/syntax/syntax.html>. Syntax. 25 de febrero de 1997.
2. <http://www.scism.sbu.ac.uk/inmandw/tutorials/nlp/index.html>. Natural Language Processing. 2002.
3. <http://www.monografias.com/trabajos5/proco/proco.shtml>. Procesamiento Computacional del Lenguaje Natural. Consultado en Agosto de 2004.
4. <http://www2.ing.puc.cl/~jabaier/iic2612/intro.pdf>. ¿Qué es Inteligencia Artificial? Jorge Baier Aranda. Consultado en Agosto de 2004.
5. http://club.telepolis.com/ohcop/clase1_fasta_ppt. Introducción a la Inteligencia Artificial. DeAnn Thomson. Noviembre de 2001.
6. <http://www.dgbiblio.unam.mx/servicios/dgb/publicdgb/bole/fulltext/volIV3/traduccion.htm>. La Traducción Automática. Jesús Valdez. UNAM, México D. F. 1989.
7. <http://elies.rediris.es/elies9/>. DISEÑO E IMPLEMENTACIÓN DE UN LEXICÓN COMPUTACIONAL PARA LEXICOGRAFÍA Y TRADUCCIÓN AUTOMÁTICA. Antonio Moreno Ortis, Facultad de Filosofía y Letras, Universidad de Málaga, España. 2000.

APÉNDICE

```
.....  
;  
.*                SECCIÓN DE PALABRAS RESERVADAS                *  
.....  
;
```

```
(defacts hechos  
(linea)  
(linea_aux)  
)
```

```
(deftemplate WORD  
(slot linea)  
(slot orden)  
)
```

```
(deftemplate ORDEN  
(slot ordenar)  
)
```

```
(defglobal  
?*ordenar* = 0  
?*orden_oracion* = 0)
```

;indica en que parte de la oración se encuentra sujeto o predicado. 1 = sujeto, 2 = predicado

```
(defclass ORDEN_ORACION (is-a USER)  
(pattern-match reactive)  
(role concrete)  
(slot parte_oracion (create-accessor write)(pattern-match reactive))  
)
```

```
(defclass FIN_ARCHIVO (is-a USER)
```

```
(pattern-match reactive)
(role concrete)
(slot fin_archivo (create-accessor write)(pattern-match reactive))
)
```

```
;-----CLASES-----
```

```
(defclass ORACION (is-a USER)
(role concrete)
(pattern-match reactive)
(multislot sustantivo_sujeto (create-accessor write))
(slot verbo_aux (create-accessor write))
(slot verbo_modal (create-accessor write))
(multislot verbo (create-accessor write))
(multislot adverbio (create-accessor write))
(slot condicional (create-accessor write))
(multislot sustantivo_predicado (create-accessor write))
(slot neg_verbo (create-accessor write))
(slot neg_sustantivo_sujeto (create-accessor write))
(slot neg_sustantivo_predicado (create-accessor write))
)
```

```
(defclass PALABRA (is-a USER)
(role abstract)
(multislot palabra)
)
```

```
(defclass SUSTANTIVO (is-a PALABRA)
(pattern-match reactive)
(role concrete)
(slot palabra (create-accessor write))
)
```

```
(defclass PREPOSICION (is-a PALABRA)
(pattern-match reactive)
(role concrete)
```

```
(multislot palabra (create-accessor write))  
)
```

```
(defclass CONJUNCIONES (is-a PALABRA)  
  (pattern-match reactive)  
  (role concrete)  
  (multislot palabra (create-accessor write))  
)
```

```
(defclass ADVERBIO (is-a PALABRA)  
  (pattern-match reactive)  
  (role concrete)  
  (slot palabra (create-accessor write))  
  (multislot tipo (create-accessor write))  
)
```

```
(defclass ARTICULO (is-a PALABRA)  
  (pattern-match reactive)  
  (role concrete)  
  (multislot palabra (create-accessor write))  
  (slot tipo (create-accessor write))  
)
```

```
(defclass DETERMINANTE (is-a PALABRA)  
  (pattern-match reactive)  
  (role concrete)  
  (multislot palabra (create-accessor write))  
  (slot tipo (create-accessor write))  
)
```

```
(defclass VERBO (is-a PALABRA)  
  (role abstract)  
  (slot palabra))
```

```
(defclass VERBO_REGULAR (is-a VERBO)
```

```
(role concrete)
(pattern-match reactive)
(slot palabra (create-accessor write))
(slot patron (create-accessor write))
)
```

```
(defclass VERBO_IRREGULAR (is-a VERBO)
(role concrete)
(pattern-match reactive)
(slot palabra (create-accessor write))
(multislot presente (create-accessor write))
(multislot pasado (create-accessor write))
(multislot futuro (create-accessor write)))
```

```
;-----FIN CLASES-----
```

```
(definstances INSTANCIAS
(FRASE of ORACION);Instancia de inicio para la oración que se está formando
)
```

```
;-----FIN OBJETOS-----
```

```
.*****
;
;*          FIN DE PALABRAS RESERVADAS          *
.*****
;
```

```
;EXPLICACIÓN DE inicio:
```

```
;esta es la regla inicial en la que se realiza la lectura del archivo
```

```
(defrule inicio
```

```
=>
```

```
(load-instances instancias.txt)
(open "c:\\propuesta.txt" archivo "r")
(bind ?linea (readline archivo))
(printout t "Leyendo archivo..." crlf)
(while (not(eq EOF ?linea))
(bind ?*ordenar* (+ ?*ordenar* 1))
```

```
(assert (WORD (linea (lowercase ?linea))(orden ?*ordenar*)))
(bind ?linea (readline archivo))
)
(close archivo)
(assert (ORDEN (ordenar 1)))
(make-instance FIN of FIN_ARCHIVO (fin_archivo 1))
(printout t "Se leyo el archivo" crlf)
)
```

```
(defrule finalizar_reglas
  ?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
  ?y <- (ORDEN (ordenar ?x))
=>
  (if (> ?x ?*ordenar*) then
    (printout t "Encontro fin de archivo" crlf)
    (save-instances frase.txt visible ORACION)
  )
)
```

```
(defrule reconoce_articulo
  (declare (salience 30))
  ?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
  ?b <- (object (is-a ARTICULO)(palabra ?x))
  ?c <- (WORD (linea ?y)(orden ?z))
  ?d <- (ORDEN (ordenar ?o))
  (test (eq ?z ?o))
  (test(= (str-compare ?x ?y) 0))
=>
  (modify ?d (ordenar (+ ?o 1)))
  (printout t "Encontro articulo " ?y ", " ?x crlf)
  (retract ?c)
)
```

```
(defrule reconoce_preposicion
  ?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
  ?b <- (object (is-a PREPOSICION)(palabra ?x))
  ?c <- (WORD (linea ?y)(orden ?z))
  ?d <- (ORDEN (ordenar ?o))
  (test (eq ?z ?o))
  (test(= (str-compare ?y ?x) 0))
=>
  (modify ?d (ordenar (+ ?o 1)))
  (printout t "Encontro la preposicion " ?y " , " ?x crlf)
  (retract ?c)
)

(defrule reconoce_verbo_irregular_presente
  ?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
  ?b <- (object (is-a VERBO_IRREGULAR)(presente ?x)(palabra ?p))
  ?c <- (WORD (linea ?y)(orden ?z))
  ?d <- (ORDEN (ordenar ?o))
  (test (eq ?z ?o))
  (test(neq (str-index ?x ?y) FALSE))
=>
  (bind ?*orden_oracion* 1)
  (printout t "Encontro verbo irregular presente " ?y " , " ?x crlf)
  (modify ?d (ordenar (+ ?o 1)))
  (slot-insert$ FRASE verbo 30 ?p)
  (retract ?c)
)

(defrule reconoce_verbo_irregular_pasado
  ?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
  ?b <- (object (is-a VERBO_IRREGULAR)(pasado ?x)(palabra ?p))
  ?c <- (WORD (linea ?y)(orden ?z))
  ?d <- (ORDEN (ordenar ?o))
```



```
(test (eq ?z ?o))
(test(neq (str-index ?y ?x) FALSE))
=>
(bind ?*orden_oracion* 1)
(printout t "Encontro verbo irregular presente " ?y ", " ?x crlf)
(modify ?d (ordenar (+ ?o 1)))
(slot-insert$ FRASE verbo 30 ?p)
(retract ?c)
)

(defrule reconoce_verbo_irregular_futuro
  ?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
  ?b <- (object (is-a VERBO_IRREGULAR)(futuro ?x)(palabra ?p))
  ?c <- (WORD (linea ?y)(orden ?z))
  ?d <- (ORDEN (ordenar ?o))
  (test (eq ?z ?o))
  (test(neq (str-index ?x ?y) FALSE))
=>
(bind ?*orden_oracion* 1)
(printout t "Encontro verbo irregular presente " ?y ", " ?x crlf)
(modify ?d (ordenar (+ ?o 1)))
(slot-insert$ FRASE verbo 30 ?p)
(retract ?c)
)

(defrule reconoce_sustantivo
  (declare (salience 10))
  ?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
  ?b <- (object (is-a SUSTANTIVO)(palabra ?x))
  ?c <- (WORD (linea ?y)(orden ?z))
  ?d <- (ORDEN (ordenar ?o))
  (test (eq ?z ?o))
  (test(neq (str-index ?x ?y) FALSE))
=>
(printout t "Encontro sustantivo " ?y ", " ?x crlf)
```

```
(modify ?d (ordenar (+ ?o 1)))
(if (eq ?*orden_oracion* 0) then
    (slot-insert$ FRASE sustantivo_sujeto 30 ?y)
else
    (slot-insert$ FRASE sustantivo_predicado 30 ?y)
)
(retract ?c)
)

(defrule reconoce_adverbio
  (declare (salience 10))
  ?a <- (object (is-a FIN_ARCHIVO)(fin_archivo 1))
  ?b <- (object (is-a ADVERBIO)(palabra ?x)(tipo negacion))
  ?c <- (WORD (linea ?y)(orden ?z))
  ?d <- (ORDEN (ordenar ?o))
  (test (eq ?z ?o))
  (test(= (str-compare ?x ?y) 0))
=>
  (printout t "Encontro adverbio " ?y " , " ?x crlf)
  (modify ?d (ordenar (+ ?o 1)))
  (if (eq ?*orden_oracion* 0) then
      (send [FRASE] put-neg_sustantivo_sujeto ?y)
  else
      (send [FRASE] put-neg_sustantivo_predicado ?y)
  )
  (retract ?c)
)

(reset)
(run)
```