# PROCESAMIENTO DE LENGUAJE NATURAL CON TRATAMIENTO DE AMBIGÜEDAD PARA LA RECUPERACIÓN DE INFORMACIÓN EN UNA BASE DE DATOS RELACIONAL


CARLOS ALBERTO GUZMÁN MURILLO

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
CALI
2010

# PROCESAMIENTO DE LENGUAJE NATURAL CON TRATAMIENTO DE AMBIGÜEDAD PARA LA RECUPERACIÓN DE INFORMACIÓN EN UNA BASE DE DATOS RELACIONAL


CARLOS ALBERTO GUZMÁN MURILLO
caralbgm@gmail.com

TESIS DE GRADO PARA OBTENER EL TÍTULO DE
INGENIERO DE SISTEMAS

MARTHA MILLÁN GONZÁLEZ, Ph.D.
DIRECTORA
millan@eisc.univalle.edu.co

RAUL ERNESTO GUTIERREZ, M.Sc.
CO-DIRECTOR
raulgdp@eisc.univalle.edu.co

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
CALI
2010

Trabajo de grado sustentado por

Carlos Alberto Guzmán Murillo

como requisito parcial para obtener el titulo de

Ingeniero de Sistemas

Aceptada por la Escuela de Ingeniería de Sistemas y Computación

| | |
|---|---|
| Martha Elena Millán González, Ph.D. | Raúl Gutiérrez, M.Sc. |
| Directora | Codirector |

Jurado

Junio de 2010

Quiero agradecer a las personas que han dado mucho si no es todo por permitirme llegar hasta aquí, mis padres. A los amigos cercanos en los que encontraba palabras de ánimo y a aquellos que directa o indirectamente me motivaban a continuar. A la niña de mi corazón que me dio fuerza y apoyo en el transcurso de elaboración de este trabajo. A la directora y el director por la confianza, guía y respaldo brindados. A todos los docentes que en el transcurso de toda la carrera me llevaron por el camino del aprendizaje, conocimiento empleado en este trabajo. A las profesoras de ciencias del lenguaje, que me dieron a conocer mucho sobre lingüística. Y por último pero no por eso menos importante, al Señor. A todos muchas gracias.

## CONTENIDO

## TABLAS



## FIGURAS

## RESUMEN

En este trabajo se propone un modelo para el procesamiento de lenguaje natural y la resolución de ambigüedades en consultas en bases de datos relacionales. El procesamiento de lenguaje natural se basa en el análisis lingüístico de la consulta. La resolución de ambigüedades se aborda con el análisis lingüístico y el uso de ontologías. De igual manera, el modelo propone el uso de lexicones y ontologías para generar un mapeo más acertado, en el sentido de obtener los datos correctos que se habían solicitado en la consulta(No generar consultas SQL que no recuperan los datos deseados). Así mismo, se describe una implementación del modelo. El prototipo resultante fue probado con la base de datos de pruebas PUBS.



**Palabras clave:** Procesamiento de lenguaje natural, PLN, análisis lingüístico, , ambigüedad, bases de datos, SQL, interfaz natural, ontología, lexicón.

## INTRODUCCIÓN

Muchos problemas de investigación abiertos existen aun en el área de procesamiento de lenguaje natural [WIK01][LEI03]. Aunque se han obtenido resultados satisfactorios en temas como la ambigüedad, la atención en esta área de investigación se está centrando, últimamente, en la utilización del procesamiento de lenguaje natural en la recuperación de información, clasificación de textos y manejo de bases de datos con interfaz natural, entre otros. Todo lo anterior se desarrolla en dominios de conocimiento aún pequeños debido al volumen y variedad de información que hoy se maneja.

Por otra parte, grandes cantidades de información se gestionan y almacenan mediante sistemas de gestión de bases de datos (SGBD). La interfaz que estos sistemas ofrecen a los usuarios para la recuperación de la información, se basa en el uso del lenguaje SQL. SQL no es un lenguaje natural para las personas, pero con él se especifican las consultas y las distintas operaciones que se pueden hacer sobre los SGBD.

Una importante tendencia con relación a la recuperación de datos, es ofrecer interfaces naturales con procesamiento de lenguaje natural para recuperar datos desde el SGBD.

En este trabajo se propone el desarrollo de un prototipo para la recuperación de información, con una interfaz en lenguaje natural, que permite la interacción con un SGBD de forma natural y tratando el problema de la ambigüedad que suele surgir en este tipo de interfaces.

En la siguiente sección se describe el problema objeto de estudio y los objetivos propuestos.

## PLANTEAMIENTO DEL PROBLEMA

Los SGBD Relacionales (SGBDR) gestionan y almacenan los datos en bases de datos bajo el enfoque relacional. Los datos se almacenan en tablas relacionadas entre sí. El lenguaje SQL, ofrece distintas operaciones sobre las bases de datos y sobre las tablas, que van desde crearlas y actualizarlas hasta eliminarlas. Cada operación se especifica por medio de una sentencia SQL, la cual cambia dependiendo del esquema relacional en el que se trabaja y la acción que se va a realizar.

Se han desarrollado aplicaciones apoyadas en distintos SGBD que ofrecen interfaces gráficas. Con estas interfaces se puede acceder y visualizar la estructura de las bases de datos, alguna información de sus componentes y realizar consultas básicas de los datos guardados[1]. Sin embargo, estas

1 Estas aplicaciones ofrecen más funcionalidades de las mencionadas, como lo son la

aplicaciones no ofrecen una manera fácil de realizar consultas complejas. Las consultas complejas involucran relaciones entre tablas, funciones de agregación y subconsultas para acceder a los datos guardados en las tablas.

Los usuarios que no son expertos en lenguaje SQL pueden tener dificultad al usar los SGBD. También, debido al poco conocimiento del lenguaje SQL, las acciones que realizan se limitan a un conjunto de operaciones básicas, sin aprovechar toda la capacidad que puede ofrecer el lenguaje.

Una manera de lograr superar este inconveniente es que tanto los SGBD como las personas que los usan se comunicaran en el mismo lenguaje. Esto haría que la comunicación fuera más fácil y permitiría una retroalimentación más clara entre el usuario y el sistema. Una interfaz usando procesamiento de lenguaje natural le permitiría al usuario, solicitar las acciones a realizar, en el lenguaje que domina y usa todos los días.

Una solución sería desarrollar una aplicación que entienda el lenguaje natural, que permita y facilite la interacción entre el usuario y el SGBD. La aplicación actuaría como un interprete entre el usuario y el SGBD. Permitiría transformar las peticiones en lenguaje natural del usuario al lenguaje que los SGBD entienden. Esto liberaría al usuario de tener que conocer detalladamente el lenguaje SQL para hacer uso de estos sistemas. Dicha aplicación tomaría como entrada una oración dada por un usuario en su lengua natural, y construiría una sentencia SQL correspondiente a la acción que se quiere realizar en el SGBD.

## OBJETIVO GENERAL

Desarrollar y probar un prototipo para la recuperación de información en bases de datos relacionales utilizando una interfaz con PLN (Procesamiento de Lenguaje Natural) y tratamiento de ambigüedad con el uso de ontologías.

## OBJETIVOS ESPECÍFICOS

- Implementar un algoritmo de análisis sintáctico basado en gramáticas para un lenguaje natural.

- Implementar un algoritmo para el análisis semántico de las estructuras sintácticas con el uso de ontologías.

- Implementar un algoritmo para el manejo de la ambigüedad en las

---

inserción de datos, lectura y creación de scripts, entre otras. Algunos ejemplos de estas aplicaciones son: phpmyadmin, pgadmin3.

consultas con el uso de ontologías.

- Definir el SGBD a utilizar y diseñar un modelo para el manejo de consultas en una base de datos relacional de forma natural, tratando el problema de la ambigüedad.

- Diseñar la interfaz en lenguaje natural teniendo en cuenta lexicones y frases a analizar en el contexto del problema.

- Implementar un analizador lingüístico que tenga tratamiento de la ambigüedad.

- Implementar el prototipo integrando el SGBD y el analizador lingüístico.

- Desarrollar una interfaz gráfica que permita la inserción de las consultas en lenguaje natural y muestre el resultado del procesamiento de dicha oración.

- Realizar las pruebas del prototipo propuesto.

- Mostrar los resultados obtenidos.

## ORGANIZACIÓN DEL DOCUMENTO

El resto de este informe está organizado en 8 capítulos. En el capitulo 1, se presentan los principales conceptos y definiciones asociados con el problema objeto de estudio. El capitulo 2, contiene una breve descripción de proyectos similares y el estado actual del problema que se aborda en este proyecto. En el capitulo 3, se propone un modelo conceptual para tratar este problema y en el capitulo 4, se describe los aspectos más relevantes en el desarrollo e implementación del modelo propuesto. En el capitulo 5, se muestran las pruebas, resultados y análisis obtenidos al usar el prototipo con el modelo propuesto implementado. En el capitulo 6, se presentan las conclusiones del proyecto. Finalmente, se esboza el glosario con los_términos técnicos y abreviaturas que aparecen en el transcurso del presente trabajo, y se listan las referencias bibliográficas de donde surgió información que colaboró con la elaboración de este proyecto.

# 1 MARCO TEORICO

El procesamiento de lenguaje natural integra varias disciplinas de las ciencias de la computación, la lingüística y la psicología, entre otras. En este proyecto se tratan temas como las bases de datos, la ambigüedad del lenguaje, las ontologías, etc. En este capítulo se presentan los conceptos y definiciones de básicas que se relacionan con este proyecto.

## 1.1 PROCESAMIENTO DE LENGUAJE NATURAL (PLN)

El procesamiento del lenguaje natural [WIK01] es una rama de la inteligencia artificial y de la lingüística computacional, que se encarga de formular e investigar mecanismos computacionales que tengan la capacidad de entender los lenguajes naturales que las personas hablan y que permita la comunicación entre personas y computadores de manera más natural. El PLN se aplica en distintas áreas, como la recuperación de información, creación de texto etiquetado(corpus), análisis semántico de textos(entendimiento del lenguaje en textos), etc. Todos estos usos, en general, se hacen con cierto entendimiento del lenguaje natural en el que se desempeñan.

Entender un lenguaje natural, básicamente, significa conocer el lexicón del lenguaje, la estructura de relaciones entre las palabras en la oración y darle el sentido correspondiente a la oración con base en las palabras que en ésta se encuentran, su estructura y el contexto en el que aparece.

### 1.1.1 Análisis morfológico

El análisis morfológico es el primer paso en el entendimiento del lenguaje[1]. Una lengua está formada por millones de palabras, que se crean, se modifican sus significados y desaparecen a medida que el tiempo pasa. Este conjunto de palabras que conforman la lengua se le conoce con el nombre de lexicón. En el análisis morfológico se analiza la estructura de la palabra con base en el lexicón de la lengua. La estructura es la base para los posteriores análisis lingüísticos. Este análisis define características de la palabra como el sentido (definición de la palabra), la raíz(lema), la categoría gramatical a la que pertenece, el género, la cardinalidad(número) y en el caso de verbos aparecen adicionalmente la conjugación, la persona, el caso y el aspecto.

1 No se considera el análisis fonético como el primer paso, ya que este análisis se puede omitir en casos donde la entrada es textual y no auditiva.

Tabla 1. Ejemplo de características morfológicas de algunas palabras

| Palabra | Categoría gramatical | Género | Número | Persona | Tiempo |
|---|---|---|---|---|---|
| casa | Sustantivo | Femenino | Singular | - | - |
| casas | Sustantivo | Femenino | Plural | - | - |
| pequeña | Sustantivo | Femenino | Singular | - | - |
| pequeña | Adjetivo | Femenino | Singular | - | - |
| pequeñas | Sustantivo | Femenino | Plural | - | - |
| vino | Sustantivo | Masculino | Singular | - | - |
| vino | Verbo | - | Singular | 3ra | Pasado |

En este análisis es importante hacer referencia a la ambigüedad, ya que a este nivel se hace evidente este problema(Tabla 1). Una palabra puede tener más de un sentido o acepción(palabras polisémicas), lo que genera el interrogante de a cuál de estos sentidos hace referencia una palabra en determinada oración. A este tipo de ambigüedad se le conoce como ambigüedad léxica.

### 1.1.2 Análisis sintáctico

Todo lenguaje tiene una estructura que define el orden en el cual aparecen las palabras en una oración, es decir, las reglas que hacen que se construyan las oraciones que lo componen. Este conjunto de reglas con las que se realizan las oraciones se le conoce como gramática y es la que define la estructura sintáctica del lenguaje. El proceso de aprender la gramática y evaluar las oraciones con dicha gramática se conoce como análisis sintáctico.

El análisis sintáctico de un lenguaje se basa en su gramática, la cual especifica la estructura de las oraciones válidas y bien formadas que hacen parte del lenguaje. El análisis sintáctico consiste en reconocer si una oración pertenece al conjunto de oraciones válidas de un lenguaje definidas por una gramática. Adicionalmente, si es una oración válida entonces generar la estructura subyacente a dicha oración que muestre de que manera se acopla a la gramática de la lengua.

La estructura más comúnmente utilizada para representar una oración y la forma como se crea a partir de la gramática de la lengua es el árbol sintáctico. Su forma varía dependiendo del tipo de gramática y de la oración que trata de representar. Por ejemplo, dada la gramática de la figura 1, el árbol sintáctico para “*aababababab*” se puede observar en la figura *2*.

$G_1 = \{ ST , SNT , T , S_0 \}$
$ST = \{ a , b , aba \}$
$SNT = \{ O , A , B \}$
$P = \{ O = AB ,$
$A = aB ,$
$A = aba ,$
$B = Ab \}$
$S_0 = \{ O \}$

Figura 1. Ejemplo gramática

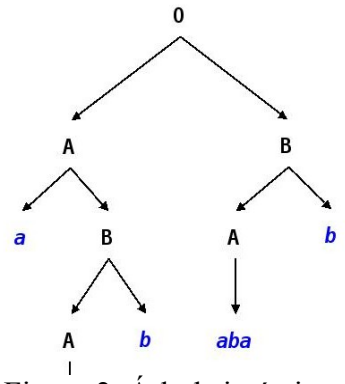


Figura 2. Árbol sintáctico

El símbolo inicial de la gramática define el nodo raíz del árbol. Cada derivación o hijo de un nodo del árbol se crea a partir de una regla de la gramática. Los nodos hoja corresponden a los producciones que no generan más símbolos *no terminales*, siendo la unión de estos nodos la oración formada. El orden en que se aplican las reglas de producción en el árbol es de izquierda a derecha, así la producción "*aababababab*" se forma como se muestra en la figura 3.

| | |
|---|---|
| O = AB | (Se aplica la regla A := aB) |
| O = aBB | (Se aplica la regla B := Ab) |
| O = aAbB | (Se aplica la regla A := aba) |
| O = aababB | (Se aplica la regla B := Ab) |
| O = aababAb | (Se aplica la regla A := aba) |
| O = aababababab | |

*Figura 3. Aplicación de las reglas de la gramática.*

Las gramáticas de las lenguas naturales se basan en las categorías gramaticales de las palabras, ya que sus reglas de producción se basan en estas categorías. Esto muestra la relación de dependencia que tiene este análisis con la morfología de las palabras(análisis morfológico).

Es preciso decir que a este nivel también se presentan problemas con la ambigüedad. Generalmente, las gramáticas de los lenguajes naturales permiten oraciones de estructura ambigua, es decir, oraciones que pueden presentar más de una estructura sintáctica válida. Estas estructuras ambiguas pueden generar confusiones para entender el sentido general de toda la oración.

### 1.1.3 Análisis semántico

Conociendo la estructura de la oración, se puede empezar a analizar el

significado que esta trae, es decir, lo que se quiere decir, la información que se quiere transmitir. A este proceso se le llama análisis semántico.

El análisis semántico de un lenguaje natural es el proceso en el cual se le asigna una interpretación o sentido a la oración como conjunto independiente de contexto. El análisis semántico se lleva a cabo después de que el análisis sintáctico se ha realizado y se ha construido una estructura sintáctica sobre la que se analiza.

Este análisis se basa, además de la estructura sintáctica de la oraciones, en las estructuras de las palabras(análisis morfológico) que constituyen la oración, el sentido, el género, el número, etc. Así, cada palabra proporciona un significado que al relacionarlo con el significados de las demás palabras conforman la idea general de la oración.

### 1.1.4 Análisis pragmático

Este análisis consiste en evaluar el sentido de la oración en un contexto. El sentido de la oración puede variar debido a las oraciones anteriores, el tema que se esta tratando, la situación, el lugar en la que se enuncia, etc. Es aquí donde el análisis pragmático actúa, este análisis también aporta a la solución de ambigüedades, descartando opciones que no son compatibles con el contexto en el que se enuncia la oración.

## 1.2 AMBIGÜEDAD

La ambigüedad en el procesamiento de lenguaje natural se da cuando a una misma oración tiene varias interpretaciones, lo cual puede conllevar a errores.

La ambigüedad se puede presentar a nivel de la palabra, la cual se conoce como ambigüedad léxica. Esta se dá cuando una palabra tiene varios sentidos y funciones morfosintácticas asociados. Este tipo de ambigüedad se puede resolver parcialmente en el análisis sintáctico de la oración, al descartar sentidos cuya categoría gramatical para la posición que ocupa la palabra dentro de la oración no concuerda con la gramática de la lengua. También al realizar el análisis semántico de la oración, la palabra se ve dentro de un contexto, lo que hace escoger los sentidos que más se adecuen con relación a las otras palabras y la idea general de la oración.

Un ejemplo de ambigüedad léxica se presenta en la siguiente oración:

- Hoy iré **solo** al museo.

En este ejemplo, la oración es ambigua, ya que se puede referir a que yo iré solamente al museo y no iré a más lugares, o también puede referirse a que iré al museo sin ninguna otra persona o compañía. Las 2 acepciones que puede

tomar la palabra *solo* produce esta ambigüedad en la oración.

- Deja la comida que **sobre sobre** la mesa de la cocina, dijo llevando el **sobre** en la mano[1].

En este caso la oración no es ambigua, pero se pueden observar los distintos significados que puede tomar la palabra *sobre*:

- Tercera persona del presente del verbo sobrar.
- Preposición: Indica que algo esta encima de otra cosa.
- Sustantivo masculino singular, objeto para llevar cartas, documentos y/o hojas.

A nivel de la oración la ambigüedad se presenta y se conoce como ambigüedad estructural. A una oración se le puede asignar más de una estructura sintáctica y por lo tanto tener más de una interpretación. Este tipo de ambigüedad se resuelve optando por la estructura que tiene un sentido coherente y se relaciona más con el contexto en el que se halla la oración.

Un ejemplo de este caso de ambigüedad es el siguiente:

- Él habló del partido de fútbol **con Juan**.

Esta oración tiene 2 estructuras sintácticas distintas que le dan a la oración 2 significados distintos. El primero se refiere a que la persona habló sobre un partido de fútbol que jugó con Juan, mientras que el segundo significado se refiere a que la persona hablo con Juan sobre un partido de fútbol.

## 1.3 SISTEMA DE GESTIÓN DE BASES DE DATOS (SGBD)

Los sistemas de gestión de bases de datos [WIK02], son aplicaciones que controlan la organización, almacenamiento, manejo y recuperación de datos en una base de datos. Además de esto, cumplen con otras funciones como mantener la consistencia e integridad de los datos, la seguridad y el respaldo, control de concurrencia, manejo de transacciones, tiempos de respuesta, etc.

## 1.4 SQL (*STRUCTURED QUERY LANGUAGE*)

SQL [WIK03], es un lenguaje declarativo creado para la recuperación y gestión de datos en los SGBD relacionales. Es un lenguaje estandarizado por la ISO y

1 Tomado de http://www.hipertext.net/web/pag277_print_minim.htm, último acceso: 11-01-10.

soportado por los SGBD, el cual permite la recuperación, inserción, actualización y eliminación de los datos en una base de datos relacional, entre otras operaciones.

## 1.5 INTERFACES CON PLN PARA BASES DE DATOS

Las interfaces con procesamiento de lenguaje natural ofrecen una manera de interactuar con el sistema más amigable y natural para los usuarios. En el dominio de las bases de datos, estas permiten interactuar con un SGBD en lenguaje natural y no en SQL. Una interfaz con lenguaje natural para bases de datos se puede ver como una capa entre el usuario y el SGBD, la cual actuaria como un "traductor" entre ambos. De esta manera, el usuario puede realizar sus consultas y/o peticiones con un lenguaje natural y la interfaz se encarga de generar su equivalente en SQL[1].

Las interfaces en lenguaje natural para bases de datos se pueden clasificar en varias áreas, según sea el enfoque de mapeo de la consulta a su correspondiente SQL [AND95]:

### 1.5.1 Concordancia de patrones

Se basan en reglas del tipo si *...X...Y...* → *Z*, es decir, si se encuentra una determinada palabra o palabras en cierto contexto y orden, entonces se asocia a una sentencia SQL determinada. Las interfaces que optan por este enfoque son muy fáciles de implementar, no requieren análisis sintáctico ni semántico de las consultas en lenguaje natural.

### 1.5.2 Con base en sintaxis

Se basa en realizar el análisis sintáctico a la consulta(usando una gramática definida o a la propia del lenguaje). Con base en la estructura sintáctica obtenida se asocia a una sentencia SQL determinada. Cada estructura sintáctica obtenida, equivale a una plantilla en lenguaje SQL. Los componentes de la estructura sintáctica son reemplazados o incluidos en la plantilla para generar la sentencia SQL. Generalmente, las plantillas o estructuras sintácticas asociadas a los arboles sintácticos son obtenidas con un previo entrenamiento sobre el dominio de la base de datos, junto a varias consultas en lenguaje natural con su respectivo SQL.

### 1.5.3 Gramáticas semánticas

1 No necesariamente la consulta se debe mapear a SQL, se puede generar en otro lenguaje, pero en el contexto de este trabajo se trabajará con SQL.

Muy similares al enfoque basado en sintaxis. La diferencia radica en que las hojas de la estructura sintáctica generadas en esta técnica pueden no ser elementos sintácticos sino de una categoría definida por la gramática semántica. Las gramáticas semánticas son pensadas para establecer restricciones y relaciones entre las palabras de la consulta, de esta manera se descartan consultas que tienen una estructura sintáctica asociada a una sentencia SQL, pero que semánticamente no tiene sentido, obteniendo un mapeo más acertado de la consulta, en el sentido de obtener los datos correctos que se solicitaron. Las gramáticas semánticas son estáticas, dependiendo del dominio de los datos contenidos en la BD.

### 1.5.4 Representación intermedia

En este enfoque, se pasa la consulta en lenguaje natural a un lenguaje lógico propio del sistema e independiente de la estructura de la base de datos. Este lenguaje expresa la consulta a un mayor nivel(conceptos de la realidad, significado de la consulta, lo que se quiere obtener) y sobre el cual el sistema puede realizar sus inferencias y análisis. Luego se convierte la consulta a su correspondiente SQL, según las inferencias y análisis realizado.

## 1.6 ONTOLOGÍA

En informática, una ontología es un modelo conceptual de un dominio de la realidad [WIK04]. Una ontología se puede considerar como un conjunto de entidades y relaciones entre ellas que se encargan de clasificar y organizar la realidad de un dominio especifico.

El uso de ontologías en sistemas computacionales es amplio y continua en crecimiento. Las ontologías son utilizadas principalmente para organizar información de un dominio, como representación del conocimiento de un dominio, para realizar inferencias, entre otros usos. Actualmente, las ontologías son muy empleadas, en campos como la web semántica, inteligencia artificial, arquitecturas informáticas, etc.

Las ontologías se componen de conceptos que clasifican la realidad y que se relacionan entre si formando una estructura jerárquica. Los conceptos pueden ser ideas, objetos, acciones, eventos, seres vivos, etc. cada uno de estos a su vez se dividen en otros conceptos y estos a su vez en otros hasta llegar a un nivel de especificación más detallado.

En las ontologías se puede emplear distintos tipos de relaciones entre los conceptos, como las relaciones de generalización y especificación, relaciones de composición, relaciones de uso, relaciones de similitud, etc. Son estas relaciones la que dan la organización a la ontología.

# 2 ESTADO DEL ARTE

Distintos proyectos y herramientas similares se han desarrollado para la interpretación de consultas hechas en lenguaje natural para la recuperación de información en bases de datos. Cada una siguiendo uno de los enfoques ya mencionados o experimentando con nuevas formas de tratar el entendimiento del lenguaje natural, a continuación se presentan algunos de estos proyectos.

*SAVVY* [AND95] es una interfaz de lenguaje natural en inglés para consultar una bases de datos creada por *Excalibur Technologies Corp* para computadores Apple. Se basa en el enfoque de concordancia de patrones. Define relaciones entre palabras y la base de datos. Cuando se ingresa la consulta en lenguaje natural, el sistema toma las palabras y las asocia a acciones ya configuradas en el sistema. Este sistema, y en general el enfoque, no tuvo gran aceptación, debido a la poca efectividad que tenía al procesar una consulta, pues el procesamiento de la consulta es muy básico y limitado, y las respuestas frecuentemente no eran las deseadas.

Un proyecto basado en sintaxis es *MaNaLa* [GIO2008]. En este proyecto se entrena el sistema con un conjunto de consultas en lenguaje natural con sus correspondientes sentencias en SQL, a partir de los cuales se generan los árboles sintácticos y relacionales respectivos que luego se usan para procesar una nueva consulta. Al ingresar una consulta, el sistema asocia su estructura sintáctica a una estructura relacional, con base en el entrenamiento previo, para generar su respuesta. Asociar la estructura sintáctica de la consulta con otra previa se realiza mediante la similitud con los arboles creados en el entrenamiento, teniendo en cuenta el árbol en su totalidad como los subárboles interiores, puesto que un subárbol de la consulta se puede asociar a otro árbol completo. Luego de tener la relación con otro árbol ya conocido, se emplea el árbol relacional de dicho árbol, el cual representa la consulta en SQL, para generar la respuesta.

*LUNAR* [AND95] es un proyecto creado para facilitarle a los científicos las consultas a una base sobre minerales y rocas encontradas en la superficie lunar. *LUNAR* se basa en el análisis sintáctico sobre las consultas en inglés para crear la sentencia SQL asociada. Aunque el proyecto se creó con un fin especifico, la base de datos y el sistema de procesamiento de las consultas son independientes, lo que permite adaptarse a otra base de datos.

Proyectos como *PLANES* [AND95] y *LADDER* [HEN77] se basaron en gramáticas semánticas (sección 1.5.3). Estos sistemas unían el análisis sintáctico y el análisis semántico para transformar la consulta hecha en lenguaje natural(inglés) a SQL. Una gramática semántica se define y se aplica

a las hojas del árbol sintáctico creado al realizar el análisis sintáctico. Esto permite imponer restricciones entre las palabras y tener una transformación semánticamente más acertada de la consulta en lenguaje natural a SQL.
Estos sistemas tienen la desventaja de ser difícilmente portables, debido a que para cada dominio de la base de datos se debe crear una gramática semántica correspondiente, lo cual ha llevado al desuso de estos sistemas y su enfoque.

En la Universidad de Edinburgh (UK) extienden *Masque* [AND92] (Sistema de respuestas modular para consultas en inglés), una interfaz para realizar consultas en lenguaje natural a bases de datos de Prolog usando Prolog. La extensión hecha en *Masque/SQL* [AND93] permite interactuar con motores de bases de datos relacionales que usan SQL, siguiendo el mismo enfoque usado en *Masque*. El sistema convierte las consultas ingresadas en inglés en una representación interna del sistema sobre la cual aplica inferencias lógicas con ayuda del motor de Prolog. Posteriormente, se transforma la estructura interna a consultas SQL. En primer instancia, se debe acceder al módulo de configuración de dominio, en el cual se describen las entidades del mundo a la cual se refiere la base de datos. Con ayuda de una jerarquía en forma de árbol, con relaciones *es-un*, se definen en términos de predicados lógicos los significados de las palabras. Al final, los predicados obtenidos se relacionan con tablas, vistas, atributos y/o sentencias SQL. Masque/SQL convierte las consultas ingresadas en ingles a consultas de Prolog llamadas LQL (*Logical Query Language*). Este lenguaje interno del sistema es un conjunto de predicados lógicos definidos previamente para cada palabra propia del dominio de la base de datos sobre el cual el motor de Prolog trabaja. Luego pasan por un algoritmo de transformación incluido en el sistema que transforma la consulta en Prolog a una consulta en SQL para ser ejecutada en el SGBD en el que se encuentra la base de datos.

*EDITE* [REI97] es una interfaz en lenguaje natural que se integró con un sistema de administración de recursos turísticos en Portugal. Este sistema es multimodal, combina interfaces gráficas y de procesamiento de lenguaje natural para la interacción con el usuario en 4 diferentes idiomas (inglés, portugués, francés y español). *EDITE* se basa en representación intermedia, convierte las consultas ingresadas por el usuario en un lenguaje de representación interno del sistema. Este lenguaje expresa el sentido de la consulta en conceptos de alto nivel, independientes de la base de datos, y sobre el cual aplica inferencias y restricciones impuestas por el dominio de la base de datos. Posteriormente, se transforma la representación interna en una consulta SQL equivalente. El sistema se compone de 2 grandes módulos: el módulo lingüístico y el módulo de transformación. En el primero, se realiza el análisis morfológico, sintáctico y semántico de la consulta. Esto produce las representaciones en el lenguaje interno del sistema, las cuales son validadas por el submódulo conceptual, en donde se realizan las inferencias y restricciones propias del dominio. El módulo de transformación, se encarga de tomar la consulta en lenguaje interno y mapearlo a SQL, con base en unas reglas de predicados ya establecidas para

el dominio de la base de datos.

Por otra parte, en [PAZ05] se describe un proyecto en el cual las consultas se realizan en español. El sistema es independiente del dominio de la base de datos. Se autoconfigura con base en los meta-datos [WK09], comentarios y descripciones que tienen los elementos (tablas, columnas, referencias, etc) de la base de datos. El diccionario del dominio se construye con base en la información extraída de la base de datos y en un diccionario de palabras de la lengua. En el diccionario también se guardan las asociaciones entre palabras y los nombres y/o descripciones de los elementos de la base de datos. Al ingresar una consulta al sistema, se realiza el análisis morfológico, sintáctico y semántico, donde se identifican la información léxica, sintáctica y semántica respectivamente para cada palabra. Luego, se divide la consulta en 2 partes, la sección de consulta (*SELECT*) y la sección de condiciones (*WHERE*). Posteriormente, para cada sección se identifican las tablas y columnas según las palabras y conexiones(se tienen en cuenta palabras auxiliares, como preposiciones y artículos) que hay entre ellas y se construye un grafo relacional. En este grafo los nodos corresponden a las tablas identificadas en la consulta y los arcos son las relaciones entre las tablas, las condiciones son etiquetadas en los nodos según las columnas que involucran y las condiciones que relacionan 2 tablas se etiquetan en los arcos que unen sus nodos correspondientes. Generalmente se obtiene un grafo conectado, el cual es fácilmente mapeado a SQL. Se pueden obtener grafos desconectados (donde hay elementos o conjunto de elementos aislados) debido a una mala consulta o a que hay condiciones implícitas que no se han tenido en cuenta. En la mayoría de ocasiones, estas condiciones implícitas se refieren a *joins* entre tablas, las cuales se resuelven con una heurística para hallar el camino que las une, y así obtener un grafo conectado.

*PRECISE* [POP03] es un proyecto que usa concordancia de patrones y análisis sintáctico, pero con una novedad en la unión de estos, el uso de grafos dirigidos. *PRECISE* realiza una reducción del problema de *tokenizar* y asociar las palabras de la consulta a los elementos de la base de datos, a un problema de grafos, el cual resuelven con el algoritmo del máximo flujo. El sistema se divide en varios módulos que se encargan de realizar el proceso de crear la sentencia SQL asociada a la consulta en inglés. Primero, toma las palabras de la consulta y las asocia a todos los elementos (tablas, atributos y valores) de la base de datos con los que son compatibles. Luego, construye un grafo dirigido con estas relaciones y aplica el algoritmo de máximo flujo para obtener el mapeo correcto de las palabras a los elementos de la base de datos. Posteriormente, el módulo generador de SQL se encarga de tomar la transformación hecha y construir el SQL correspondiente. En caso de encontrar ambigüedades que pasaron el módulo de *matching*(donde se realiza la reducción y solución del grafo) y las restricciones impuestas por el análisis sintáctico hecho a la consulta, es el usuario quien ofrece la interpretación correcta que se le debe dar a la consulta.

De acuerdo con los autores, el sistema sobre *preguntas semánticamente tratables* es completo, en el sentido de que puede tratar con cualquier pregunta de este tipo, y confiable, en el sentido de que las respuestas producidas son las correctas [POP03] [POP04]. Este tipo de preguntas cumplen varias restricciones, como que las palabras de la pregunta estén contenidas en el lexicón y estas a su vez se asocien con elementos de la base de datos, que la pregunta en general tenga cierta estructura (siempre debe ser una pregunta), etc.

Un proyecto reciente, que se basa en en gramáticas semánticas, aprendizaje de máquinas y representación interna es *C-PHRASE* [MIN09]. Este proyecto se enfoca en asociar palabras y frases a elementos de la base de datos. A través del módulo de configuración, los usuarios crean estas asociaciones. Cuando la consulta en inglés se ingresa, se realiza el análisis morfológico, las correcciones sobre las palabras que estén mal escritas y se descartan las desconocidas. Posteriormente, se realiza el análisis sintáctico (usando gramáticas libres de contexto y el calculo lambda), donde se asocian las palabras y frases con los elementos de la base de datos. Luego de todo este proceso, se genera una o más representaciones de la consulta en términos del *cálculo relacional orientado a de tuplas de codd*, las cuales posteriormente son mapeadas a SQL. *C-PHRASE* incluye un generador de lenguaje natural, el cual transforma una consulta en el lenguaje interno del sistema a una representación en lenguaje natural. Esto se usa para corregir ambigüedades con ayuda del usuario, de esta manera, si una consulta tiene una o más interpretaciones, estas son mostradas al usuario para que sea él quien decida la interpretación correcta.

En [TSE08], se propone una interfaz en lenguaje natural para consultas a bases de datos, basadas en diagramas de clase de la base de datos. Primero, se crea un diagrama de clases asociado a la estructura de la base de datos, al cual le añaden información semántica acerca de las palabras que se pueden asociar a cada clase atributo o relación del diagrama. Al ingresar la consulta en inglés, esta pasa por el análisis sintáctico, donde se definen los roles y dependencias de las palabras. Posteriormente, basado en el árbol de dependencia de la oración, ésta se transforma al lenguaje interno, que es una extensión de la consulta al diagrama de clases de la base de datos. Luego, la consulta en lenguaje interno es convertida a una expresión equivalente en álgebra relacional, desde la cual se puede realizar una fácil transformación a SQL.

En el centro nacional de investigación y desarrollo tecnológico de México, están realizando una interfaz en lenguaje natural para realizar consultas a bases de datos relacionales en Español [ZAR09]. Dicho proyecto se basa en el proceso lingüístico y en el uso de ontologías para el procesamiento de la consulta. En primer instancia, el sistema toma la consulta y le aplica el análisis léxico,

sintáctico y semántico, en donde intervienen tanto la ontología como el diccionario de palabras de la lengua. Posteriormente, la consulta es pasada al generador de código, en donde se construye la sentencia SQL y se envía a la base de datos. El sistema usa *DAML*[1], un lenguaje para la gestión de ontologías, y el diccionario de palabras de *WordNet* [MIL95] [MIL09]. Además de esto, el sistema sera una aplicación web, que permita el uso multiplataforma del sistema. La entrada de las consultas al sistema se realizará por voz.

1 DARPA Agent Markup Language. http://www.daml.org/

# 3 MODELO CONCEPTUAL

En este capítulo se describe el modelo conceptual propuesto para el procesamiento de las consultas en lenguaje natural a una base de datos. En primer lugar, se describe, de manera general, la estructura y funcionamiento del modelo. Se finaliza presentando el modelo conceptual dividido en módulos, donde se describen más detalladamente sus funcionalidades y características.

## 3.1 DESCRIPCIÓN GENERAL

El modelo conceptual propuesto se basa en el análisis lingüístico de la consulta, su interpretación interna y el uso de ontologías. Básicamente, como se muestra en la figura 4, el sistema realiza, en primera instancia, un análisis lingüístico de la consulta en lenguaje natural. Al finalizar este análisis se obtiene una representación interna de lo que el usuario desea consultar en la base de datos. Sobre esta estructura, se realizan los procesos de análisis e inferencia, para obtener la forma lógica de la consulta en la base de datos y finalmente esta se trasforma a SQL para que se pueda ejecutar en el SGBD.

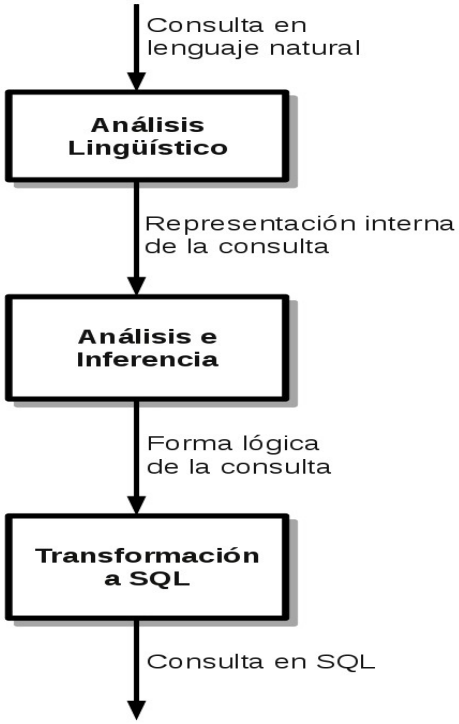


Figura 4. Procesamiento general de la consulta

El modelo propuesto(figura 5), se enfoca en el análisis lingüístico de la consulta y el uso de ontologías. El análisis lingüístico facilita la construcción de una estructura semántica de la oración que representa lo que el usuario quiere extraer de la base de datos. En este proceso se emplea una ontología, además de los recursos necesarios para realizar el propio análisis lingüístico, como lo son el lexicón y la gramática de la lengua. Posteriormente, se lleva a cabo el proceso de análisis e inferencia, en el cual se analiza la manera de obtener los datos referentes a la consulta. Este proceso se basa principalmente en las funcionalidades del sistema y en el esquema relacional de la base de datos. El resultado final es la expresión del álgebra relacional para la consulta. Seguidamente, la expresión algebraica se convierte en una expresión SQL equivalente, para ser ejecutada en el SGBD.

Es importante resaltar dos elementos muy importantes en este modelo: el lexicón y la ontología.

### 3.1.1 El lexicón

El diccionario de palabras del modelo se compone de tres sub-lexicones, el primero es el lexicón general de la lengua, el segundo es el lexicón del dominio de SQL y bases de datos en general, y por último, el lexicón propio del dominio de la base de datos(figura 6).

En el lexicón general, se guardan todas las palabras que es capaz de reconocer el sistema, incluyendo las del dominio de SQL y las del dominio de la base de datos. En el lexicón del domino de SQL y bases de datos, se guardan aquellas palabras que se usan en el contexto de consultas a bases de datos y/o se refieren a características propias de construcciones en SQL. Por último, en el lexicón del dominio se guardan únicamente aquellas palabras propias del dominio de la base de datos. Esta división se hace para darle prioridad a aquellos sentidos que están mas relacionados con la base de datos a la hora de interpretar la consulta.

El sistema usa el lexicón semántico *WordNet* en español provisto en *FreeLing*[1] [MIL95] para representar el lexicón general de la lengua, este recurso lingüístico de distribución libre, contiene gran cantidad de palabras en español agrupadas en conjuntos de sinónimos llamados *synsets*, que a su vez se relacionan con otros dándole su característica de lexicón semántico.

1 http://www.lsi.upc.edu/~nlp/freeling/

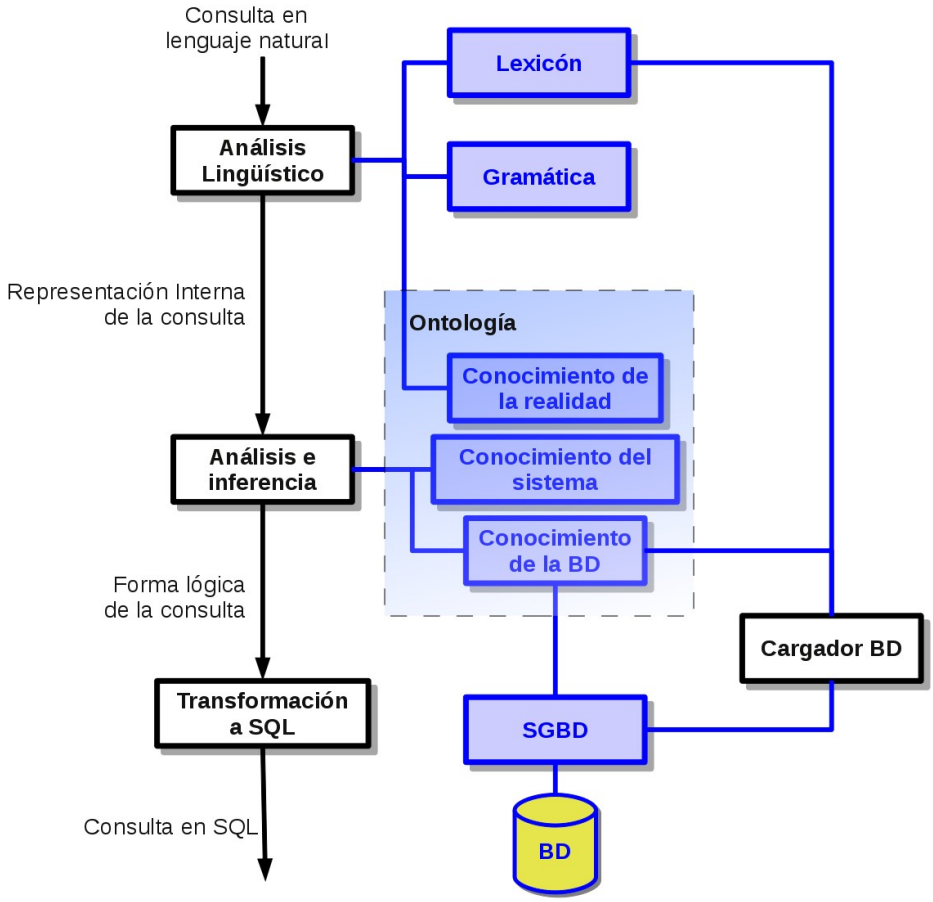


Figura 5. Modelo conceptual

### 3.1.2 La ontología

La ontología propuesta por el modelo se puede dividir en tres grupos de conocimiento: el conocimiento de la realidad en general, el conocimiento del sistema y sus funcionalidades, y el conocimiento del dominio de la base de datos. Cada uno de estos conocimientos se puede ver como una ontología que los representa y su unión conforman la ontología total del sistema(figura 7).

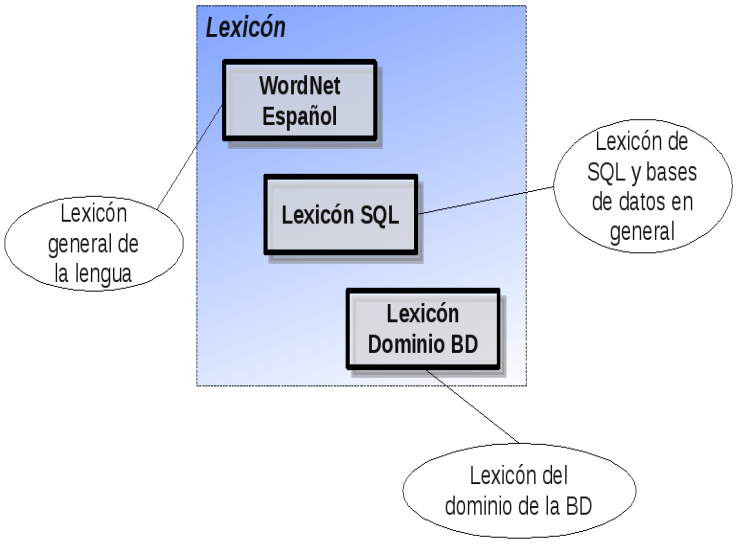


Figura 6. Lexicón

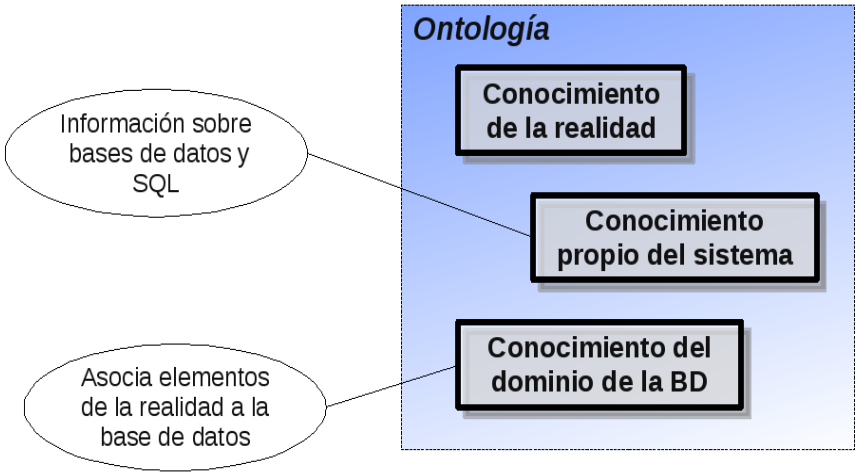


Figura 7. Ontología

La ontología de la realidad en general, como su nombre lo indica, representa el conocimiento de toda la realidad, independiente del sistema y del dominio de la base de datos. El sistema usa la ontología provista por el lexicón semántico *WordNet* [MIL95] [MIL09], en donde la realidad esta jerárquicamente organizada en grupos de sentidos(*synsets*). De esta manera, cada aspecto de la realidad se puede clasificar en alguno de los grupos y relacionarse con otros

grupos a través de los diversos tipos de relaciones que maneja esta ontología.

El conocimiento del sistema se representa de igual manera por una ontología. En esta se encuentran las funcionalidades y características propias del modelo propuesto, es el conocimiento de si mismo y lo que es capaz de realizar. Esta ontología también contiene información de las bases de datos en general y el lenguaje SQL. De esta manera, el sistema conoce la manera de seleccionar datos, unir tablas, tiene conocimiento sobre llaves primarias llaves, foráneas y sabe como realizar una función de agregación o un conteo de registros, etc.

El conocimiento del dominio de la base de datos, es una ontología que contiene la información de la base de datos sobre la que se van a realizar las consultas. Esta ontología enlaza directamente el sistema con la información contenida en la base de datos y como esta se organiza. Los sentidos que se asocian con elementos de la base de datos se encuentran en esta ontología.

### 3.1.3 Representación interna del sistema

La representación interna del sistema describe la consulta en términos de alto nivel, es decir, la semántica de la consulta, lo que se quiere transmitir en la consulta.

La representación interna del sistema se ha definido como el árbol lingüístico de la oración. Este árbol representa la estructura sintáctica de la consulta en lenguaje natural. También esta etiquetado semánticamente, con los sentidos de cada una de las palabras y constituyentes de la estructura sintáctica. Además de esto, contiene información semántica adicional, sobre los roles semánticos de los verbos que contiene la oración y los argumentos que los representan.

### 3.1.4 Forma lógica de la consulta

La forma lógica de la consulta es un lenguaje formal interno en que el sistema representa la forma de obtener los datos pedidos de la base de datos. En este modelo se utilizará el álgebra relacional [WIK06] como forma lógica de la consulta, ya que esta define la forma en que se debe proceder para obtener los datos de un esquema relacional. Además, su transformación a SQL es bastante sencilla, y sobretodo facilita la posibilidad de extender el modelo para generar la consulta en otros lenguajes distintos al SQL.

## 3.2 MÓDULOS

El modelo conceptual propuesto se divide en varios módulos (desglosados a continuación), según su funcionalidad, los cuales a su vez se componen de submódulos que pueden estar presentes también en otros módulo distintos.

### 3.2.1 Módulo de Análisis Lingüístico

En este módulo se procesa, inicialmente, la consulta ingresada en lenguaje natural. La función de este módulo es convertir la consulta ingresada en una estructura interna del sistema, sobre la cual se puedan aplicar el resto de análisis. Para esto, se realizan el análisis morfológico, sintáctico, semántico y pragmático sobre la consulta.

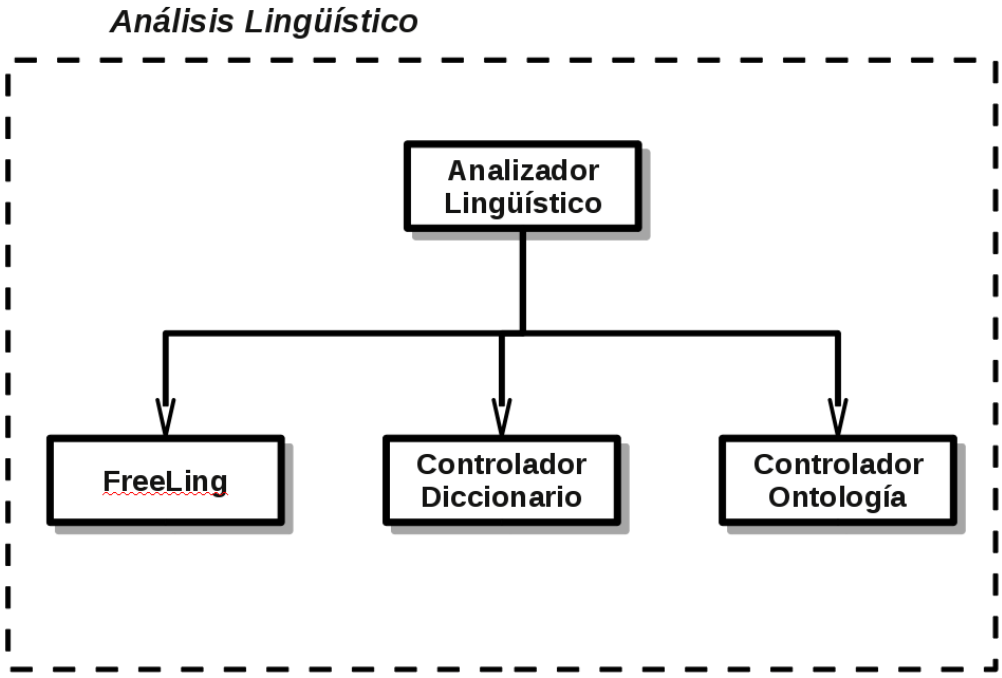


Figura 8. Módulo análisis lingüístico

Como se muestra en la figura 8, este módulo usa otros submódulos para cumplir con la totalidad de su tarea.

**Analizador Lingüístico:** Es el encargado de controlar todo el proceso y el resto de los submódulos. Primero, se realiza el análisis morfológico por medio de *FreeLing* (sección 3.2.1). Con base en el análisis realizado, se lleva a cabo el análisis sintáctico y semántico, usando al algoritmo *chart parser*. Con esto se obtiene una estructura en árbol, con información sintáctica y semántica de la consulta. Finalmente, se realiza el análisis pragmático con base en esta estructura para obtener la consulta en el lenguaje interno del sistema.

El análisis sintáctico toma una gramática de la lengua y construye el árbol sintáctico de la oración. El algoritmo y gramática han sido extendidos para

soportar el uso de *características* [ALL95]. Las características definen valores para atributos en los símbolos terminales y no terminales. Con el uso de características es posible tener en cuenta fenómenos de concordancia [WIK08] al realizar el análisis sintáctico, sin tener que extender ampliamente la gramática.

En este análisis se logran resolver ambigüedades léxicas. Se descartan aquellas acepciones de una palabra que no correspondan con la categoría gramatical que debe asumir dada su posición en la oración y la gramática de la lengua. De igual manera, se descartan aquellas acepciones que no cumplan con las restricciones de concordancia impuestas por las características adicionadas en la gramática.

En el análisis semántico, se agregan al árbol sintáctico información semántica independiente del contexto. En este análisis se resuelven ambigüedades léxicas y estructurales, descartando aquellos sentidos que no tienen cavidad con los sentidos de las demás palabras. Para la resolución de la ambigüedad en este análisis se usan 2 técnicas: análisis de roles semánticos y desambigüación con el uso de la ontología [ALL95].

La primer estrategia de desambigüación usada en el análisis semántico se basa en los roles semánticos de los verbos presentes en la consulta. Por cada verbo, es posible hallar el papel semántico que tienen sus argumentos [WIK07] [WAG02]. Estos papeles semánticos definen aspectos como el agente, el lugar, el afectado, etc. del verbo. Cada papel semántico tiene características que deben cumplir el argumento que lo representa. Por ejemplo, el papel semántico de agente lo debe representar un ser animado, y el papel semántico de lugar lo debe representar un constituyente que designe un lugar.

Esta estrategia extrae el predicado y sus argumentos de la oración. Luego, se obtienen los papeles semánticos asociados al predicado. Posteriormente, se determinan puntajes para las distintas combinaciones que se pueden establecer para asociar los argumentos del predicado a sus papeles semánticos. El sistema de puntaje para las combinaciones, da prioridad a aquellas asignaciones donde se cumplen el mayor número de restricciones impuestas por los sentidos de los roles.

Con esta estrategia se logra resolver tanto ambigüedades estructurales como léxicas. Se pueden descartar aquellas interpretaciones de la oración que no cumplan con el número de argumentos (generalmente definidos por la estructura sintáctica de la oración) correspondiente a los papeles semánticos del predicado. En tanto a la ambigüedad léxica, una vez definidos los papeles semánticos y sus correspondientes argumentos, se descartan aquellos sentidos de los argumentos que no son compatibles con la definición del papel semántico que representan.

Después de aplicar la desambigüación con base en los papeles semánticos, si la ambigüedad continua, se pasa a aplicar la desambigüación con ayuda de la ontología.

En la desambigüación con ontología, solo se resuelven ambigüedades léxicas. En esta estrategia, se aplica el algoritmo UKB [AGI09]. En general, el algoritmo crea un subgrafo de la ontología con los sentidos de la palabra que se va a desambigüar, los sentidos del resto de palabras de la oración y las relaciones entre todos estos sentidos. Posteriormente, se asignan valores a los sentidos de las palabras que rodean la palabra objetivo a desambigüar, y se propagan dichos valores sobre los sentidos de la palabra objetivo. Al construir el subgrafo, se tiene en cuenta la ontología a la cual pertenece la relación palabra-sentido de las palabras y sentidos que rodean la palabra a desambigüar. Dependiendo de la ontología a la que pertenezcan, se les asignan los valores iniciales para propagar. Aquellas relaciones palabra-sentido que estén en la ontología del dominio de la base de datos objetivo tendrán un mayor valor para propagar. De esta manera, se da prioridad a los sentidos más cercanos al dominio de la base de la datos sobre la cual se quiere realizar la consulta. Finalmente, si algún sentido de la palabra a desambigüar obtiene una un valor máximo de 1.0 o una diferencia mayor o igual a 0.8[1]con respecto a los sentidos competidores, se opta por este sentido y se descartan el resto.

En el análisis pragmático el cual se lleva a cabo junto al análisis semántico. Al aplicar el algoritmo UKB para desambigüación con ayuda de la ontología, los sentidos de la sentencia previa más reciente, también son cargados en el subgrafo. De esta manera, se tiene en cuenta el contexto del discurso en el proceso de desambigüación. El sistema de puntuación en este análisis, da prioridad a aquellas palabras cuyos sentidos obtuvieron puntajes altos(entre más alto sea el puntaje del sentido, más relación tiene con el resto de palabras) y al grado de ambigüedad resultante, entre mayor sea la diferencia entre los puntajes de los sentidos, significa que la palabra tiende más al sentido con puntaje más alto.

Finalmente, para cada interpretación obtenida se suman los puntajes obtenidos en cada análisis. Con base en este puntaje, dando prioridad a aquellas con puntaje más alto, el módulo de análisis e inferencia generara las sentencias SQL correspondientes.

Al finalizar el análisis lingüístico, se obtiene la representación interna de la consulta ingresada(o varias interpretaciones en caso no poder resolver la

1 Este valor es arbitrario. Se utilizó como 0.8 después de realizar varias pruebas y notar que los sentidos no relevantes en el contexto, tenían valores muy bajos.

ambigüedad). Sobre esta representación interna trabaja el módulo de análisis e inferencia.

**FreeLing:** El análisis morfológico se lleva a cabo usando los analizadores presentes en las librerías de *FreeLing* [ATS06].

En el análisis morfológico de la consulta, ésta se divide en palabras que se etiquetan con sus características léxicas como el género, cardinalidad, categoría gramatical, etc. Algunas palabras son etiquetadas como números, fechas o signos de puntuación.

En este análisis, también se etiquetan los sentidos y las probabilidades de cada acepción de las palabras de la consulta. Los sentidos son identificadores referentes a la ontología de *WordNet*. Las probabilidades asignadas a las palabras, se basan en el análisis de corpus etiquetados, de los cuales se extraen las ocurrencias de los sentidos de las palabras.

**Controlador Diccionario:** Este submódulo se encarga de ofrecer una interfaz para agregar nuevas palabras al lexicón del sistema. Con esta interfaz, es posible agregar palabras y sentidos asociados al dominio de la base de datos objetivo sobre la cual se realizarán las consultas.

Después de ingresar la palabra o adicionar un nuevo sentido a una palabra, son realizados los cambios pertinentes en la ontología a través del submódulo Controlador Ontología.

**Controlador Ontología:** El controlador de ontología se encarga de ofrecer una interfaz para acceder a las relaciones entre dos conceptos de la realidad, así como reestructurar su jerarquía con la inserción de nuevos elementos y la modificación semántica de otros.

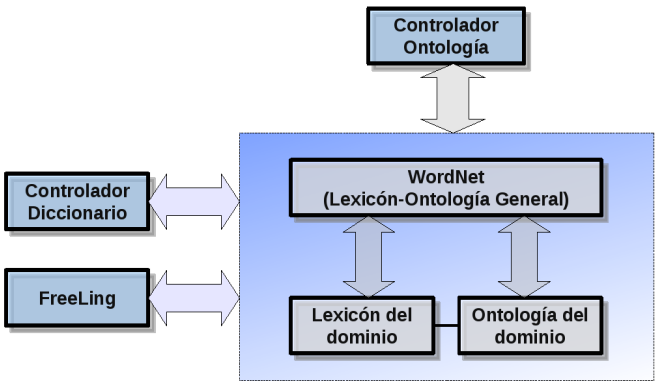


Figura 9. Lexicones y ontologías

Al momento de ingresar una nueva palabra, además de ser ingresada en el lexicón del sistema, se adiciona a la ontología. De igual manera que con el lexicón, el sentido se agrega a una de las subontologías del sistema.

### 3.2.2 Módulo de análisis e inferencia

Este módulo toma la salida producida por el análisis lingüístico, la representación interna de la consulta, y crea su respectiva forma lógica. Este módulo se encarga de obtener la estructura de la consulta para obtener los datos de la base de datos. Se usa tanto la información de la estructura de la base de datos, como la ontología del sistema.

Teniendo la representación semántica de la consulta, se realiza el proceso de transformación a la estructura de la base de datos. Posteriormente, se crea la expresión en álgebra relacional para extraer los datos.

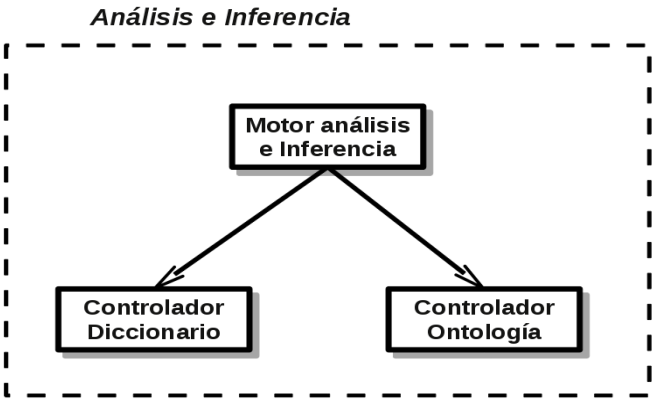


Figura 10. Módulo de análisis e inferencia

Como se muestra en la figura 10, este módulo contiene y usa otros submódulos:

**Motor análisis e inferencia:** En este submódulo se realiza el proceso en general de todo el módulo. Se toma la representación interna de la consulta, y con base en sus características sintácticas y semánticas se realiza la transformación a la forma lógica.

Las tablas de las bases de datos están relacionadas entre si a través de las llaves primarias y foráneas. Estas relaciones representan distintos conceptos y relaciones entre los conceptos de la realidad. Por ejemplo, una tabla *estudiante* (contiene información sobre estudiantes) se relaciona con una tabla *universidad* (contiene información sobre universidades). Esta relación significa que un estudiante *estudia* en una universidad. En la estructura de la base de

datos esta relación se representa con una llave foránea o por medio de una tabla puente. Como el ejemplo anterior, muchas relaciones entre los conceptos de la realidad se representan en la estructura de una base de datos (por medio de llaves foráneas a primarias o de tablas puente). También es posible que exista más de una forma de relacionar 2 tablas: una relación directa (a través de una llave foránea) y una o más de una relación indirecta (formando caminos a través de otras tablas). Así como las tablas se relacionan entre si, los atributos de una tabla se pueden ver como una relación *parte-de* de la tabla que los contiene. Por ejemplo, en la tabla estudiantes, se puede encontrar el atributo *nombre*. Conceptualmente, se puede ver que el *nombre* es parte del *estudiante*. En casos donde el atributo es una llave foránea a otra tabla, no necesariamente implica una relación *parte-de*. Siguiendo el ejemplo anterior, entre estudiantes y universidades, si la tabla de estudiantes contiene el identificador de la universidad como atributo (una llave foránea a la tabla *universidad*) esto no representa que la universidad es parte del estudiante. En este caso, el atributo representa la relación de *estudiar*. Un estudiante *estudia* en una universidad. Con base en esto, las consultas relacionan conceptos de la realidad e imponen restricciones sobre las características de los conceptos. La dificultad se encuentra en identificar los conceptos de la consulta y las relaciones que los unen correctamente.

La ontología también ayuda a establecer las relaciones involucradas en la consulta. Las relaciones semánticas que contienen los sentidos de la ontología permiten una identificación más acertada de las tablas y atributos. Por ejemplo, la palabra *estudiar*, indica que *alguien* estudia *algo* en algún *lugar*. El sentido de esta palabra tiene asociados tres papeles semánticos. El *tema* (lo que se estudia), el *agente* (quien estudia) y el *lugar* (donde se estudia). En la consulta *¿Dónde estudia Juan?* seria posible identificar los conceptos correctos (bajo el contexto de la consulta) de las palabras con base en los conceptos relacionados a los papeles semánticos. El agente representa una persona que realiza la acción. En esta consulta, el agente está representado por *Juan*. El lugar está representado por el pronombre interrogativo *dónde*. Por lo tanto, la consulta implica una relación entre una persona y un lugar. Posteriormente, con base en el conocimiento de la base de datos, se busca una relación de este tipo. Con la ontología, se pueden clasificar los sentidos que representan los elementos de la base de datos. De esta manera, se puede saber que elementos representan una persona y cuales un lugar.

Teniendo la representación interna de la consulta, se busca identificar las tablas, atributos y relaciones asociados a los sentidos contenidos en la consulta. Para identificar las tablas y atributos se busca en la ontología del dominio de la base de datos los sentidos de los constituyentes de la consulta. Indirectamente, también es posible identificar tablas por medio de sus atributos y valores. Una vez identificado un atributo, es posible conocer la tabla a la que pertenece. Las relaciones entre las tablas, se hallan con base en las relaciones sintácticas y semánticas de la representación interna de la consulta. Las tablas

más cercanas sintáctico/semánticamente se asocian primero, y la unión de estas a su vez se unen con otras tablas o uniones.

Los sentidos de las palabras de la consulta definen acciones a realizar sobre sus argumentos o se pueden referir a elementos de la base de datos. En el primer caso, los sentidos actúan como predicados, cuyos argumentos están definidos por la estructura sintáctica en la que se encuentran. Conceptos como el de la palabra *mayor*, imponen restricciones a sus argumentos. Este tipo de palabras no se refieren a un elemento de la base de datos directamente, pero si puede afectarlos. Las restricciones impuestas por estos predicados se establecen con base en la propia definición del concepto, sus relaciones con otros conceptos y el tipo de estas relaciones.

Una tabla o atributo de la base de datos, puede estar representada por más de un sentido en la ontología. Por ejemplo, un atributo con nombre *nombre_autor*, puede estar asociado al concepto de *nombre* y al concepto de *autor*. De la misma manera, un sentido puede representar más de un elemento en la base de datos. Por lo tanto, se puede dar el caso de tener asociado a un sentido de la consulta más de un elemento de la base de datos. Con la ayuda de la ontología se intenta resolver esta ambigüedad. También, la relación con el resto de sentidos de la consulta aporta a la solución de esta ambigüedad(algoritmo UKB [AGI09]).

Aquellos sentidos a los que no se le puede obtener un elemento de la base de datos equivalente, generalmente representan valores de los atributos o formas de obtener la consulta(ordenamiento, agrupaciones, etc). En el caso de los valores, estos establecen restricciones sobre los atributos de las tablas.

Una consulta a una base de datos consta de 4 secciones principales. La primera son los datos que se quieren obtener. La segunda, es el conjunto de tablas fuente para obtener los datos. La tercera, es el conjunto de restricciones que se quiere imponer sobre las tablas fuentes, para obtener ciertos datos. La cuarta, es la forma (orden, agrupamiento, etc) en que se quieren obtener los datos.

Aquellos atributos identificados en la consulta y que no están restringidos a un valor, equivalen a los datos que se quieren extraer con la consulta. En caso de no encontrarse con atributos sin valores, se extraen todos los datos de la tabla o tablas relacionadas con la representación interna de la consulta.

Las tablas identificadas en la consulta, corresponden a la segunda parte de la consulta a una base de datos. Estas tablas son las fuentes para los datos que se desean extraer.

Los atributos asociados a valores, imponen las restricciones de la consulta a la base de datos. Estos establecen que valores para ciertos atributos deben

cumplir los datos que se quieren extraer.

Por último, los modificadores son aquellos sentidos que no se pudieron asociar ni a una tabla ni a un atributo o valor de la base de datos. Estos modificadores indican la forma de obtener los datos. Pueden referirse a ordenamientos, a agrupamientos, a limitación de cantidad de respuestas, etc.

El sistema de puntuación para las formas lógicas de la consulta, da prioridad a aquellas formas lógicas que relacionan tablas más cercanas y a aquellas donde se especifica las columnas y explícitamente sus respectivas tablas.

En caso de que no se halla resuelto toda la ambigüedad de la consulta en el análisis lingüístico, este módulo aplica el mismo procedimiento sobre cada una de las interpretaciones. Aquellas interpretaciones a las cuales no se les puede generar una forma lógica válida, se descartan. Las interpretaciones son tratadas y mostradas dando prioridad a aquellas que obtuvieron los puntajes más altos en el análisis lingüístico.

**Controlador diccionario:** Este submódulo es empleado para buscar palabras en los lexicones del sistema y obtener su respectivo sentido.

**Controlador ontología:** Este submódulo es usado para buscar sentidos en la ontología propia del dominio de la base de datos y en la ontología del conocimiento del sistema. Con este submódulo se pueden obtener los elementos a los que se refiere el sentido de una palabra en el contexto de la base de datos.

### 3.2.3 Módulo de transformación a SQL

Este módulo se encarga de transformar la forma lógica de la consulta en su sentencia SQL equivalente. Como se menciono anteriormente, este modelo representa la forma lógica de la consulta con álgebra relacional.

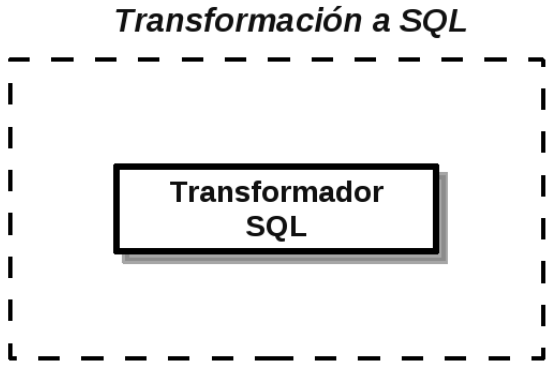


Figura 11. Módulo de transformación a SQL

Como se muestra en la figura 11, este módulo contiene un único submódulo encargado de transformar una expresión en álgebra relacional a una expresión SQL equivalente.

Básicamente, toma las 4 partes que componen la consulta mencionadas anteriormente, y las inscribe en la sintaxis propia del SQL.

**Interfaz SGBD:** Este submódulo se encarga de interactuar con el SGBD. Por medio de la interfaz que ofrece para la comunicación con el SGBD, se ejecuta la consulta en SQL y se obtiene la respuesta que el SGBD retorna.

### 3.2.4 Módulo cargador BD

Este módulo se encarga de inicializar el sistema para la base de datos objetivo. En este módulo se carga la información semántica de la base de datos al sistema. La información semántica se extrae de los meta-datos contenidos en la base de datos. Los nombres de las tablas y atributos son procesados e incluidos tanto al lexicón y la ontología del sistema. Los nombres de los elementos de la base de datos objetivo se buscan en la ontología del sistema y se asocian sus respectivos conceptos a los elementos de la base de datos. A los conceptos asociados a cada tabla y sus respectivos atributos, se le aplica la desambigüación UKB para descartar aquellos sentidos que no se relacionan con el dominio de la base de datos. Los conceptos asociadas a las tablas vecinas son utilizados como el contexto de los conceptos a desambiguar.

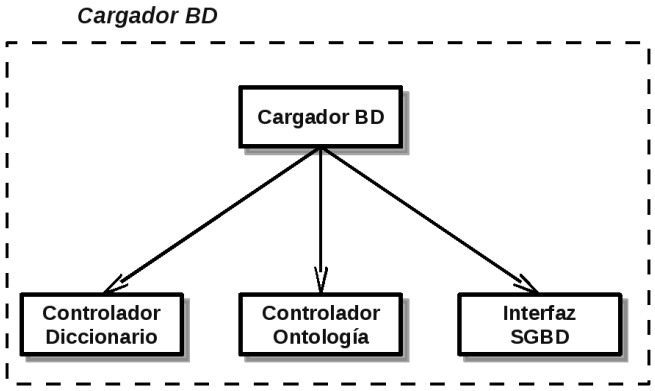


Figura 12. Módulo cargador BD

Como se ve en la figura 12, este módulo usa los siguientes submódulos:

**Cargador BD:** Este submódulo lleva a cabo el proceso en general de la carga de información semántica de la base de datos al sistema.

En primer instancia se leen los meta-datos de la base de datos a través del módulo interfaz SGGB. Los nombres de las tablas y atributos son procesados para hallar palabras contenidas en el lexicón y asociar sus sentidos a estos elementos de la base de datos. Si no es posible asociar una palabra con sentido a un atributo o tabla, se le pregunta al usuario por una o varias palabras que representen la información contenida en la tabla o atributo.

Teniendo los sentidos asociados para cada tabla y atributo, se aplica la desambigüación con ayuda de la ontología(UKB) para descartar aquellos sentidos que no son cercanos al dominio general de la base de datos. En esta desambigüación, se usan las palabras identificadas en la base de datos y sus sentidos asociados. Finalmente, con los sentidos establecidos para los atributos y tablas de la base de datos, se agregan tanto al lexicón como a la ontología del dominio de la base de datos. Adicionalmente, se agregan relaciones directas entre los sentidos de los atributos a los sentidos de sus respectivas tablas en la ontología del dominio de la base de datos.

**Controlador Diccionario:** Con este submódulo se consultan y se agregan las palabras calve y representativas de la base de datos. Estas palabras son agregadas en el lexicón propio del conocimiento de la base de datos.

**Controlador Ontología:** Con ayuda de este submódulo, se agregan a la ontología los sentidos de las palabras extraídas de los nombres de los elementos de la base de datos.

**Interfaz SGBD:** Este submódulo se emplea para la lectura de los meta-datos de la base de datos. Con este submódulo se extraen los nombres de las tablas y atributos. Se extraen las relaciones entre las tablas (llaves foráneas) y los atributos identificadores de cada tabla (llaves primarias).

# 4 DESARROLLO Y DETALLES DE IMPLEMENTACIÓN DEL MODELO

Se desarrolló un prototipo del modelo propuesto para la plataforma GNU/Linux. El prototipo fue desarrollado usando el lenguaje de programación C++ debido a los altos costos computacionales que requieren el análisis lingüístico y los procesos de desambigüación. El prototipo usa el gestor de bases de datos MySQL.

Para la ontología y lexicón del sistema se usó WordNet 3.0 y su respectivo mapeo al español[1]. La ontología del WordNet se manejó en una base de datos de MySQL[2]. La ontología del sistema fue ampliada con los papeles semánticos de los verbos provista en VerbNet 3.1[3]. De esta manera, se asociaron a los conceptos de los verbos sus respectivos papeles semánticos.

En el proceso de análisis e inferencia, se agregó en la implementación acciones para el procesamiento de predicados de algunos conceptos. Estos predicados se definen con base en los sentidos de las palabras. La estructura sintáctica de la consulta especifica los argumentos del predicado. Las palabras asociadas con algunos de los predicados agregadas son *seleccionar*, *listar*, *dar*, *mayor*, *menor*, *igual*, etc. Este tipo de palabras requieren de acciones adicionales sobre sus argumentos. La adición de más palabras clave y/o palabras que requieren de acciones adicionales es un proceso dispendioso, por esta razón no se llevó a cabo. Con este conjunto de pocas palabras, se logra mostrar el funcionamiento del modelo propuesto.

En el caso de las palabras *listar* y *dar*, la acción a realizar consiste en crear la sentencia SQL a partir de sus argumentos e ignorándose como elemento a incluir en la estructura SQL de la consulta (no se usa como un posible valor para un atributo). Las funciones de las palabras *mayor*, *menor* e *igual* actúan como operadores entre atributos y valores en la sentencia SQL. De igual manera, se pueden definir acciones para palabras cuyos conceptos indiquen la manera de obtener los datos, como lo es la palabra *ordenar*.

1 Wordnet mappings, NLP Research Group. http://www.lsi.upc.edu/~nlp/web/index.php?option=com_content&task=view&id=21&Itemid=57 . Último acceso: 03-06-10
2 WordNet SQL Builder. http://wnsqlbuilder.sourceforge.net/ . Último acceso: 03-06-10
3 VerbNet. http://verbs.colorado.edu/~mpalmer/projects/verbnet.html . Último acceso; 03-06-10

# 5 PRUEBAS Y ANALISIS DE RESULTADOS

Al sistema se ingresa la consulta a la base de datos en lenguaje natural(español), y este devuelve la sentencia SQL correspondiente a la base de datos objetivo ya definida. Según como se configure, el sistema devuelve una o varias sentencia SQL asociadas a la consulta ingresada. De esta manera, por cada consulta ingresada se puede obtener una sola sentencia SQL o un conjunto de sentencias SQL ordenadas por el puntaje otorgado durante todo el proceso de transformación.

Por ejemplo, al ingresar la consulta "*Dame los títulos de los libros*", el sistema devuelve las siguientes sentencias SQL:

*SELECT libro.titulo FROM libro*
*SELECT libro.titulo,libro.libro_id FROM libro*
*SELECT libro.titulo,autor_libro.libro_id FROM libro,autor_libro WHERE*
*libro.libro_id=autor_libro.libro_id*
*...*

El prototipo fue probado con la base de datos de prueba PUBS. Esta base de datos modela un conjunto de librerías. Contiene información de libros: títulos, autores, editoriales, etc. También contiene información sobre los empleados de las librerías y sus cargos y sobre las editoriales.

Primero se probaron los procesos de desambigüación sobre las consultas. Por cada consulta se tomó la interpretación que obtuvo el mejor puntaje al aplicarle los procesos de desambigüación y se verificó si era correcta de acuerdo a la consulta ingresada. Por ejemplo, la consulta "*Dame la dirección y la ciudad de las editoriales*", se analiza de la siguiente manera: Primero se aplica el proceso de desambigüación léxica en base a sus probabilidades. Por ejemplo, la palabra "la" tiene dos acepciones, como determinante y como nombre común(la nota musical). En este caso, se toma la acepción referente al determinante ya que su probabilidad de aparición es más frecuente que la acepción referente a la nota musical. Luego se obtiene el árbol sintáctico con mejor puntaje, en este caso el árbol de la figura 13. Finalmente, se desambiguan sus sentidos con el algoritmo UKB. En la tabla 2 se muestran los resultados.

| | **Correctas** | **Incorrectas** | **Total** | **%** |
|---|---|---|---|---|
| Desambigüación. Léxica | 42 | 5 | 47 | 89,36 |
| Desambigüación sintáctica | 32 | 15 | 47 | 68,09 |
| Desambigüación Semántica | 19 | 28 | 47 | 40,43 |

Tabla 2. Resultados desambigüación.

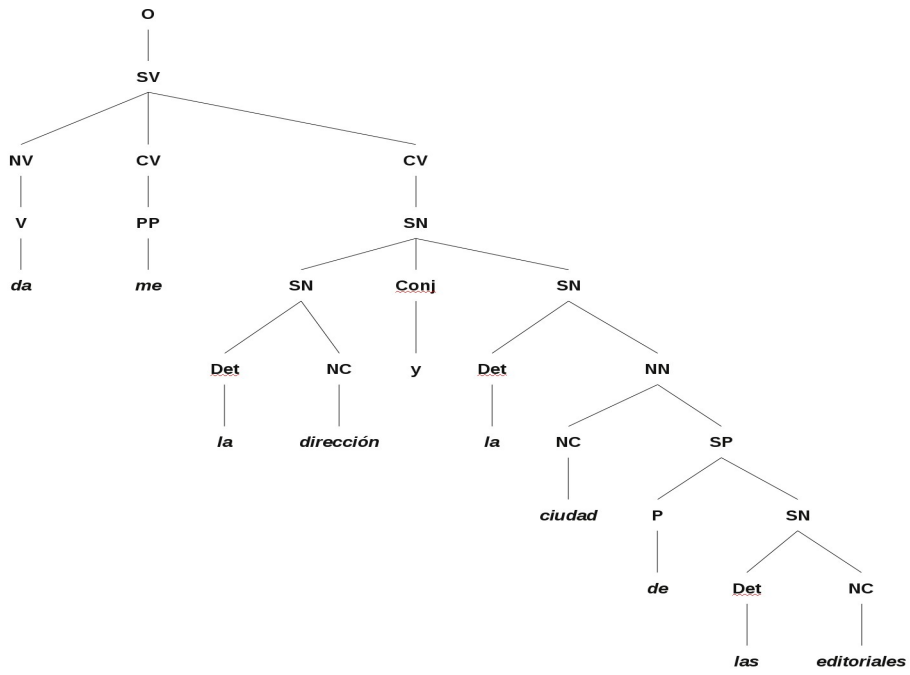


Figura 13. Árbol sintáctico simple

Los resultados se clasificaron según el tipo de ambigüedad resuelta. La ambigüedad léxica se refiere a las palabras cuya ambigüedad es de tipo gramatical. La ambigüedad sintáctica se refiere a las distintas estructuras sintácticas a las que se puede asociar la consulta. Finalmente, la ambigüedad semántica se refiere a los distintos sentidos que tenían las palabras de la consulta.

Debido en parte a la poca efectividad en el proceso de desambigüación, al aplicar la transformación a SQL de las interpretación obtenida, no se logró obtener una sentencia SQL correcta para la consulta ingresada. Pero al permitir cierto nivel de ambigüedad(se permitió tener 3 sentidos máximo por palabra) y evaluando todas las múltiples sentencias SQL generadas(las cuales en su mayoría no correspondían a la consulta), en el 38% de los casos se genero la sentencia SQL correcta para la consulta ingresada. En la tabla 3 se muestran los resultados.

| Consultas | Correctas | Incorrectas | Total | % |
|---|---|---|---|---|
| Todas | 18 | 29 | 47 | 38,30% |
| Sin ambigüedad sintáctica | 15 | 7 | 22 | 68,18% |
| Sin ambigüedad. semántica | 14 | 7 | 21 | 66,67% |
| Sin ambigüedad sintáctica ni semántica | 13 | 5 | 18 | 72,22% |

Tabla 3. Resultados de la transformación de las consultas a sentencias SQL.

En todos los casos, la consulta ingresada generó más de una sentencia SQL sintácticamente válida (una sentencia que se puede ejecutar en la base de datos objetivo sin producir error), pero que no correspondían a la consulta ingresada. En el 38% de los casos, se encontró la sentencia SQL correspondiente. Las sentencias SQL generadas son ordenadas y presentadas según su puntaje obtenido durante el proceso de desambigüación y transformación. En la tabla 4 se muestran los resultados promedio cuando se generan sentencias SQL para las consultas. También se muestra el promedio de la posición en que la sentencia correcta se encontraba de acuerdo a su puntaje.

| Consultas | Promedio Generadas | Promedio posición puntaje |
|---|---|---|
| Todas | 65,81 | 2,53 |
| Sin ambigüedad sintáctica | 49,94 | 2,8 |
| Sin ambigüedad semántica | 59,52 | 2,43 |
| Sin ambigüedad sintáctica ni semántica | 44,17 | 2,54 |

Tabla 4. Resultados sentencias SQL generadas por consulta

Como se puede observar los resultados obtenidos no son satisfactorios. Un análisis de los resultados evidenciaron varios problemas. La mayoría de sentencias SQL creadas no corresponden a lo que se solicita en la consulta. Los elementos de la base de datos objetivo asociados a los conceptos de las palabras de la consulta no son correctos. Muchas sentencias SQL contienen palabras que se usan como valores pero que en la consulta no representan un valor.

Algunos ejemplos evidenciando estos errores se muestran a continuación:

1. Dame la ciudad de la editorial "New Moon Books".
   SELECT editorial.ciudad FROM editorial
   SELECT editorial.ciudad FROM editorial WHERE
   editorial.nombre_editorial='New_Moon_Books'
   SELECT editorial.ciudad FROM editorial WHERE
   editorial.editorial_id='New_Moon_Books'

2. Obtén los números de ejemplares que tiene el libro “the_busy”.
   SELECT venta.numero_venta FROM venta,libro WHERE venta.libro_id=libro.libro_id
   AND libro.libro_id='the_busy'
   SELECT venta.numero_venta,libro.libro_id FROM venta,libro WHERE
   venta.libro_id=libro.libro_id

3. Dame la ciudad en donde se encuentra el autor Johnson White.
   SELECT autor.ciudad FROM autor WHERE autor.autor_nombre='Johnson'
   SELECT autor.ciudad FROM autor WHERE autor.autor_apellido='Johnson'

4. Dame la ciudad a la que pertenece el código postal 89076

```
SELECT * FROM tienda WHERE tienda.ciudad='89076' AND tienda.tienda_id='postal'
SELECT * FROM tienda WHERE tienda.ciudad='89076' AND
        tienda.codigo_postal='postal'
```

En el primer ejemplo, las palabras de la consulta se podían identificar con múltiples elementos de la base de datos. La palabra editorial, se asociaba con la tabla *editorial*, con la columna *nombre_editorial*, la columna *editorial_id*, etc. En el segundo ejemplo, la palabra *números* se asoció incorrectamente con la columna *numero* de la tabla venta. En el tercer ejemplo, en cada sentencia, la palabra *autor* solo se asocia a un elemento de la base de datos. Por lo tanto, una de las dos partes del autor(nombre y apellido), que se usan como valores de columnas, no se usan. En el último ejemplo, se muestra como la palabra *postal*, no se une con la palabra *código* para asociarse a la columna *codigo_postal*, y por el contrario, en una de las sentencias se utiliza como valor para esta columna que se asocia a la palabra *código*.

El modelo propuesto falla, principalmente, en la resolución incompleta de ambigüedades y en los recursos utilizados. Una de las fallas se identifica en la ontología utilizada (WordNet 3.0). Esta ontología contiene un conjunto limitado de relaciones entre los conceptos. Las pocas relaciones no permiten relacionar muchos conceptos. La ambigüedad producida por los diversos elementos de la base de datos objetivo asociados a los conceptos de las palabras de la consulta, no se puede resolver completamente con asociaciones entre los conceptos de las palabras, ya que en algunos casos no existían en la ontología.

El análisis lingüístico efectuado sobre la consulta en lenguaje natural, no permite la solución completa de la posible ambigüedad presente en la consulta. En este análisis se pueden obtener distintas interpretaciones para una consulta. A pesar de los métodos de resolución de ambigüedad empleados, en muchos casos no se logra obtener una única interpretación válida y semánticamente correcta(correspondiente a la consulta ingresada). Esto se debe principalmente a las distintas acepciones de las palabras usadas en la consulta.

Dadas estas ambigüedades, se producen múltiples e incorrectas asociaciones entre las palabras y los elementos de la base de datos objetivo. En consultas como "*Dame el precio del libro con identificador de editorial 1389*", se pueden obtener asociaciones entre la palabra *identificador,* la columna *identificador_libro* de la tabla *libro* y la columna *identificador_editorial* de la tabla *editorial*.

En otros casos, se descartan las interpretaciones correctas. Esto debido a que en los procesos de desambigüación se descartan acepciones correctas de las palabras de la consulta. En el caso del algoritmo UKB, esto se produce cuando conceptos de las palabras vecinas a la palabra que se está desambigüando no

tienen asociados dichos conceptos en la ontología(más específicamente en el mapeo de los sentidos de WordNet3.0 a las palabras del español, muchas palabras no tienen sus conceptos asociados [1]). Entonces, al no estar presente aquellos conceptos que refuerzan el concepto correcto de la palabra que se esta desambigüando o al no haber relación en la ontología entre los conceptos, se puede dar el caso de obtener la acepción incorrecta de la palabra en el contexto de la consulta. Adicionalmente, este método no tiene en cuenta la estructura sintáctica de la consulta, por lo que conceptos de palabras distantes(en la estructura sintáctica) a la palabra que se esta desambigüando, tienen el mismo peso en el proceso de desambigüación.

Por lo tanto, en consultas como "*Dame el precio del libro con identificador de editorial 1389*" se pueden obtener acepciones incorrectas de las palabras. Si no se puede relacionar el concepto correcto de la palabra *editorial* con el concepto correcto de la palabra *identificador*, se puede obtener como correcta la asociación *identificador* con la columna *identificador_libro* de la tabla *libro*.

Aunque semánticamente muchas de las interpretaciones obtenidas no son válidas, no se pueden descartar. Varias de las interpretaciones que se obtienen relacionan(en la estructura sintáctica de la interpretación) incoherentemente los conceptos de las palabras. Estas interpretaciones no son posible de eliminar, debido a que no hay forma de distinguirlas de las interpretaciones que si son semánticamente válidas. A causa de las limitadas relaciones en la ontología usada, no es posible saber si la relación entre los conceptos no es válida semánticamente o si no están representadas en la ontología.

Por ejemplo, en consultas como "*Dame el precio del libro con identificador de editorial 1389*", se pueden obtener interpretaciones en donde "*identificador de editorial 1389*" afecta(o se relaciona sintácticamente) a "*precio*", aunque esta interpretación sea semánticamente incorrecta. No hay manera de establecer una distinción(sin usar el esquema de la base de datos objetivo) entre esta interpretación y la interpretación correcta, donde "*identificador de editorial 1389*" afecta a "*libro*". Aunque en este ejemplo y dada el esquema de la base de datos objetivo, a la interpretación incorrecta no se le puede asociar una forma lógica válida, existen otros casos en donde esto si se puede dar.

Adicionalmente, la desambigüación con el uso de los roles semánticos de la oración se aplica parcialmente. En primera instancia, no todos los verbos tienen identificados y relacionados sus roles semánticos en la ontología(VerbNet). En segunda instancia, no existía una definición de los diferentes tipos semánticos

1 Wordnet mappings, NLP Research Group. http://www.lsi.upc.edu/~nlp/web/index.php?option=com_content&task=view&id=21&Itemid=57 . Último acceso: 03-06-10

de los roles aplicados a los verbos en la ontología(VerbNet). Es decir, el rol *agente* asociado al verbo *estudiar*, no tienen asociado ningún sentido en la ontología, por lo que no se podía imponer como una restricción sobre los argumentos de la consulta que los representaban. Aunque se definieron los roles de una manera general (lo cual permite argumentos inválidos pero que son compatibles con el sentido asignado al rol), estos obtienen sentidos más específicos y diferentes en cada verbo. En tercera instancia, algunos roles semánticos requieren que los argumentos que lo representan en la oración tengan una estructura sintáctica especifica. Esta restricción sintáctica es dependiente del lenguaje. Al usar un recurso que no es natural del español, estas restricciones no se pueden aplicar ya que estas formas sintácticas no son necesariamente iguales en ambos lenguajes(ingles-español).

En el proceso de análisis e inferencia, varios de los conceptos asociados a las palabras de la consulta son relacionados con objetos de la base de datos objetivo, mientras que otros, que no tienen relación explicita con la base de datos objetivo, se usan como valores para los atributos nombrados en la consulta. Muchas de las formas lógicas producto de las combinaciones entre los elementos de la base de datos identificados y los posibles valores, se descartan por su mala formación. Las formas lógicas mal formadas son aquellas que tienen valores sin asociar a ningún atributo. También aquellas que relacionan tablas que en el esquema de la base de datos objetivo no se pueden relacionar. También son consideradas como formas lógicas mal formadas aquellas en donde los tipos de los valores no corresponden a un tipo compatible con la columna a la que se asocian(en este caso, se verifica que los atributos numéricos no tengan valores no numéricos). Sin embargo, muchas de las combinaciones son válidas aunque no representen semánticamente la consulta ingresada.

Otro problema identificado fue la ambigüedad presente en los sentidos asociados a los elementos de la base de datos objetivo. Por ejemplo, la palabra identificador, se puede asociar a todos los elementos que tienen este concepto asociado, como *editorial_id*, *autor_id*, *libro_id*, etc. Esta ambigüedad influye, además de la generación de la sentencia correcta con mejor puntaje, en la cantidad sentencias generadas. Todas las asociaciones posibles son combinadas entre sí con base en su estructura sintáctica, para generar las sentencias finales.

Varias de las palabras que no se relacionan explícitamente con los elementos de la base de datos objetivo, pueden indicar acciones a aplicar sobre estos elementos. Por ejemplo, la palabra *máximo*, se usa en "*¿cual es el libro con precio máximo?*" para indicar que el valor de la columna *precio* de la tabla *libro*, debe ser el mayor.

Palabras como "*contratar*" implican restricciones sobre los conceptos con los cuales se relacionan y que no están representadas en las relaciones entre

dichos conceptos. Este concepto en el contexto de la base de datos objetivo se refiere a que una editorial *contrata* a una persona. En la definición del concepto de *contratar* implica que la persona contratada se convertirá en un empleado. Este tipo de restricciones o información semántica adicional no se encuentra representada en la ontología del sistema.

En consultas como "*Dame la fecha de contratación de Pedro*", no se puede asociar el concepto de *contratar* con el concepto de la tabla *empleado*, ya que no hay una relación entre estos dos conceptos.

Palabras que hacen parte del conjunto de palabras cerradas de la lengua[1] y que no están incluidas en la ontología, también indican implícitamente acciones o relaciones entre los conceptos y los elementos de la base de datos. La preposición "*de"*, usada en la consulta "*Dame el precio del libro con identificador de editorial 1389*" indica que el concepto de la palabra *editorial* se relaciona directamente con el concepto de la palabra *libro*. La conjunción *y*, en la consulta "*Dame la dirección y la ciudad de las editoriales*" indica que la forma lógica debe incluir ambos campos relacionados con la tabla *editorial*. La preposición "*en"* indica una restricción semántica distinta a la preposición "*por"* en las siguientes consultas; "Dame la cantidad de libros vendidos en Bookbeat" y "Dame la cantidad de libros vendidos por Carson". En el primer caso, la preposición usada es "en" e indica un lugar. En el segundo caso, la preposición usada es "por" e indica una persona. De igual manera, otras palabras de este tipo tienen asociadas restricciones que influyen en la estructura semántica de la consulta.

Estas palabras que no están incluidas en la ontología, al no tener un sentido claro también afectan en la semántica de la consulta. El modelo propuesto no tiene en cuenta este aspecto y por lo tanto, no puede aprovechar esto para desambigüar la consulta y obtener la forma lógica correspondiente a la consulta.

1 Las palabras cerradas de la lengua son aquellas que cumplen funciones gramaticales. Esta clase de palabras que difícilmente cambia con el tiempo, son pocas las que cambian sus sentidos, pocas las nuevas que se agregan y pocas las que desaparecen. Esta clase de palabras se componen de las preposiciones, pronombres, artículos, conjunciones y auxiliares.

# 6 CONCLUSIONES

Aunque con este trabajo no se logró resolver la ambigüedad en las consultas a bases de datos, es posible que el enfoque tenga mejores resultados. El modelo propuesto que se enfoca principalmente en la desambigüación de la consulta con el uso de ontologías, saca provecho de las propiedades semánticas de los distintos conceptos de las palabras en la consulta. Teniendo una ontología más completa, en el sentido de tener más información semántica de los conceptos, es posible que se obtengan mejores resultados en cuanto a desambigüación de la consulta.

Una ontología que represente más relaciones entre los conceptos de la realidad, permite relacionar más conceptos de las palabras de la consulta y de una forma más directa y precisa, en el sentido del tipo de relación entre los conceptos. Con esto, en el algoritmo de desambigüación UKB, los conceptos correctos de las palabras en la consulta se retro-alimentan entre sí y de esta manera descartar los incorrectos.

También la ambigüedad producida al asociar los elementos de la base de datos a los conceptos se debe tener en cuenta. Esta ambigüedad produce la mayor parte de sentencias semánticamente incorrectas con respecto a la consulta ingresada.

Otro punto importante a concluir al finalizar este trabajo es que la desambigüación total de la consulta, no es suficiente para obtener la sentencia SQL asociada a la consulta. La definición de los conceptos puede abarcar información extra que no se representa en las relaciones con los otros conceptos. La definición del concepto de la palabra *ganar* en el contexto del *fútbol*, puede relacionar conceptos asociados a las palabras *equipo* y *gol*. Pero su propia definición implica un conjunto de condiciones adicionales. Por ejemplo, deben relacionarse 2 *equipos* y a su vez uno de ellos debe tener más *goles* que el otro.

Una ontología que contenga este tipo de información puede permitir una mejor resolución de la ambigüedad. Esto se debe a que con una ontología de esta naturaleza es posible imponer restricciones a las relaciones entre conceptos y a los conceptos relacionados. Aquellos conceptos que no cumplan con las restricciones, equivaldrían a conceptos incorrectos dentro de ese contexto y se podrían descartar. Esta ontología también debe tener esa información adicional bien representada. Este tipo de definiciones deben estar en términos de conceptos de la misma ontología y un conjunto de operaciones base. De lo contrario, interpretar la definición podría convertirse en otro problema a

solucionar.

Con este tipo de definiciones en la ontología, sería posible interpretar y obtener la forma lógica correcta de la consulta. Con definiciones de palabras como *mayor*, *total*, *distinto*, *y*, *no*, etc. que generalmente implican operaciones más avanzadas sobre los datos que se quieren obtener con la consulta, sería posible interpretar sus conceptos con base en sus definiciones y llevar a cabo las acciones adicionales necesarias sobre la forma lógica de la consulta.

De la misma manera, y dependiendo del contexto de la base de datos objetivos, se pueden establecer las definiciones de algunos conceptos que requieran de condiciones y/o acciones adicionales en la forma lógica de las consultas en las que se presentan, como era el caso del concepto de la palabra *contratar* visto en el análisis de las pruebas.

Se concluyo también que es necesario la interpretación de las palabras cerradas del lenguaje(preposiciones, pronombres, conjunciones, etc). Este tipo de palabras aunque carecen de un sentido claramente definido, como ya se mostró, influyen en la semántica de la consulta.

Finalmente, con los recursos utilizados y los métodos de desambigüación e interpretación de la consulta empleados, no fueron suficientes para obtener las sentencias SQL correctas de las consultas en lenguaje natural analizadas.

## TRABAJOS FUTUROS

Vista la posibilidad de obtener mejores resultados. Se podría seguir explorando este camino.

El tener en cuenta la estructura sintáctica de la oración en la desambigüación UKB, podría tener mejores resultados en cuanto a obtener los conceptos correctos de las palabras de la consulta y en general a la interpretación correcta de la consulta.

Aplicar los procesos de desambigüación al asociar las palabras de la consulta a los elementos de la base de datos.

Otro aspecto que mejoraría el proceso de desambigüación de las consultas es darle manejo a palabras cerradas del lenguaje como lo son el caso de las preposiciones.

Implementar el modelo sobre una ontología que contenga las definiciones de los conceptos con base en otros conceptos y sus relaciones, sería la mejor prueba a hacer para verificar si el modelo tiene un mejor desempeño para este problema.

## GLOSARIO

AMBIGÜEDAD: cualidad de tener más de un significado o sentido.

CHART PARSER: técnica empleada para realizar el análisis sintáctico de un lenguaje formal.

DAML: (*DARPA Agent Markup Language*). Lenguaje para la definición y etiquetado de ontologías.

DARPA: (*Defense Advanced Research Projects Agency*). Agencia de Proyectos de Investigación Avanzada de la Defensa de los Estados Unidos.

GRAMÁTICA: conjunto de reglas que generan las oraciones validas para una lengua.

LÉXICO: conjunto de palabras que componen una la lengua. Diccionario de una lengua.

LEXICÓN: conjunto de palabras que componen una la lengua. Diccionario de una lengua.

LQL: (*Logical Query Language*). Lenguaje lógico de consulta.

METADATO: Datos que describen otros datos [WK09].

ONTOLOGÍA: representación del conocimiento en un sistema computacional. Clasificación de la realidad en un sistema informático.

PARSER: analizador sintáctico de un lenguaje.

PLN: procesamiento de lenguaje natural.

POSTGRESQL: sistema de gestión de base de datos de distribución gratuita.

SGBD: sistema de gestión de base de datos.

SQL: (*Structured Query Language*). Lenguaje estructurado de consulta.

TOKENIZAR: Separar la consulta en palabras clave o en conjunto de palabras claves.

WORDNET: base de datos léxico semántica para el ingles.

## BIBLIOGRAFÍA

ALL95: Allen, J.. . . The Benjamin/Cummings Publishing Company, 1995. ISBN 0-8053-0334-6

AND92: Androutsopoulos, I.. Interfacing a Natural Language Front-End to a Relational Database. 1992

AND93: Androutsopoulos, I., Ritchie, G., Thanisch, P.. MASQUE/SQL: an efficient and portable natural language query interface for relational databases. 1993

AND95: Androutsopoulos, I., Ritchie, G. G., Thanisch, P.. Natural Language Interfaces to Databases - An Introduction. 1995

ATS06: Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.. FreeLing 1.3: Syntactic and semantic services in an open-source NLP library . 2006

FER08: Fernández-Montraveta, A., Vázquez, G., Fellbaum, C.. The Spanish Version of WordNet 3.0. 2008

GIO2008: Giordani, A.. Mapping Natural Language into SQL in a NLIDB. 2008

HEN77: Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., Slocum, J.. Developing a natural language interface to complex data. 1977

MIL09: Miller, G. A.. WordNet - About Us.. . 2009. http://wordnet.princeton.edu. último acceso: 11-02-10

MIL95: Miller, G. A.. WordNet: a lexical database for English. 1995

MIN09: Minock, M.. C-PHRASE: A system for building robust natural language interfaces to databases. 2009

PAZ05: Pazos R. A., Perez, J. González, J. J., Gelbukh, A., Sidorov, G., Rodríguez, M.. A Domain Independent Natural Language Interface to Databases Capable of Processing Complex Queries. 2005

POP03: Popescu, A. M., Etzioni, O., Kautz, H.. Towards a Theory of Natural Language Interfaces to Databases. 2003

REI97: Reis, P. Matias, J. Mamede, N.. Esite - A Natural Language Interface to

Databases A new dimension for an old approach. 1997

TSE08: Tseng F. S.C., Chen, C.. Enriching the class diagram concepts to capture natural language semantics for database access. 2008

WAG02: Wagner, A.. Learning Thematic Role Relations for wordnets. 2002

WIK01: . Natural language processing. Wikipedia. 2010. http://en.wikipedia.org/wiki/Natural_language_processing. último acceso: 11-01-10

WIK02: . Database management system. Wikipedia. 2009. http://en.wikipedia.org/wiki/Database_management_system. último acceso: 11-01-10

WIK03: . SQL. Wikipedia. 2010. http://en.wikipedia.org/wiki/SQL. último acceso: 11-01-10

WIK04: . Ontology. Wikipedia. 2010. http://en.wikipedia.org/wiki/Ontologies. último acceso: 11-01-10

WIK05: . First-order logic. Wikipedia. 2009. http://en.wikipedia.org/wiki/First-order_logic. último acceso: 11-01-10

WIK06: . Relational algebra. Wikipedia. 2010. http://en.wikipedia.org/wiki/Relational_algebra. último acceso: 11-01-10

WIK07: . Teoría-θ. Wikipedia. 2009. http://es.wikipedia.org/wiki/Teor%C3%ADa-%CE%B8. último acceso: 11-01-10

WIK08: . Agreement (linguistics). Wikipedia. 2009. http://en.wikipedia.org/wiki/Agreement_(linguistics). último acceso: 11-01-10

WK09: . Metadata. Wikipedia. 2010. http://en.wikipedia.org/wiki/Metadata. último acceso: 18-01-10

ZAR09: Zárate, A., Pazos, R., Gelbukh, A., Padrón, I.. A Portable Natural Language Interface for Diverse Databases Using Ontologies. 2009