



**INSTITUTO TECNOLÓGICO  
DE CD. MADERO**



**DIVISIÓN DE ESTUDIOS DE  
POSGRADO E INVESTIGACIÓN**



**“Estrategia de Mejora para la Independencia del Dominio en  
una Interfaz de Lenguaje Natural”**

**PARA OBTENER EL GRADO DE:  
MAESTRO EN CIENCIAS EN CIENCIAS DE LA COMPUTACIÓN**

**PRESENTA:  
I.S.C. ARODIT MARGARITA GONZÁLEZ MICHEL**

**DIRECTOR DE TESIS:  
DR. JUAN JAVIER GONZÁLEZ BARBOSA**

**JURADO**  
**Presidente: DR. JOSÉ ANTONIO MARTÍNEZ FLORES**  
**Secretario: DR. RODOLFO ABRAHAM PAZOS RANGEL**  
**Vocal: DR. JUAN JAVIER GONZÁLEZ BARBOSA**  
**Suplente: M.C. GUADALUPE CASTILLA VALDEZ**

"2011, Año del Turismo en México"



SUBSECRETARÍA DE EDUCACIÓN SUPERIOR  
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA  
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

Cd. Madero, Tamps; a 29 de Marzo de 2011

OFICIO No.: U5. 123/11  
AREA: DIVISIÓN DE ESTUDIOS  
DE POSGRADO E INVESTIGACIÓN  
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

**C. ARODIT MARGARITA GONZÁLEZ MICHEL  
P R E S E N T E**

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

**"ESTRATEGIA DE MEJORA PARA LA INDEPENDENCIA DEL DOMINIO EN UNA INTERFAZ DE LENGUAJE NATURAL"**

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

**ATENTAMENTE**

*"Por mi Patria y por mi Bien"*

*M. P. María Yolanda Chávez Cinco*  
**M. P. MARÍA YOLANDA CHÁVEZ CINCO**  
JEFA DE LA DIVISIÓN



**S.E.P.**  
DIVISION DE ESTUDIOS  
DE POSGRADO E  
INVESTIGACION  
I T C M

c.c.p.: Archivo  
c.c.p.: Minuta

MYCHC 'MCO' 'aygc'

# RESUMEN

En las últimas décadas, la información ha jugado un rol importante en nuestra vida cotidiana; la mayoría de las personas solicitan información antes de tomar alguna decisión importante. Actualmente las mayores fuentes de información se encuentran almacenadas dentro de bases de datos, las cuales permiten modelar la información que se encuentra en el mundo real.

Para que un usuario pueda obtener información de una base de datos necesita formular una consulta de tal manera que la computadora interprete y produzca la respuesta correcta, esta consulta deberá estar formulada en el lenguaje formal de bases de datos; desafortunadamente, sólo usuarios con experiencia pueden realizar esta clase de consultas.

La manera normal en la que las personas solicitan información es mediante consultas en lenguaje natural, pero las computadoras no pueden entender este lenguaje. Debido a esto, las Interfaces de Lenguaje Natural para Bases de Datos (ILNBDs) surgen como una alternativa a varios problemas que se presentan en los sistemas y medios para obtener información, ya que permiten al usuario acceder a información almacenada en bases de datos mediante una solicitud en lenguaje natural.

Las consultas que el usuario realiza a menudo están formuladas con información incompleta, la cual se debe determinar para poder interpretar el significado de la oración y realizar una traducción correcta al lenguaje formal.

A pesar de que desde finales de los años 60s se han implementado infinidad de ILNBDs, en esta área aún existen problemas, que hasta la fecha, las metodologías implementadas no han podido solucionar completamente. Por lo tanto, el objetivo de

este trabajo de tesis es lograr un incremento en el porcentaje de éxito en la traducción de consultas de lenguaje natural al lenguaje formal SQL (Structured Query Language). Este trabajo de tesis continúa con los esfuerzos previos realizados en la ILNBD desarrollada en el ITCM.

Existe una técnica de traducción basada en el manejo de sustantivos que involucra el tratamiento de preposiciones y conjunciones. Sin embargo el tratamiento dado a las preposiciones no contemplaba por completo los casos en los que se presentaba este problema, mediante un estudio previo de esta problemática, fue posible identificar las áreas débiles de este tratamiento.

Se diseñó e implementó una estrategia de solución factible que contempla aquellos casos que habían sido omitidos, esto permitió obtener resultados favorables incrementando el porcentaje de éxito de la interfaz. Se llevaron a cabo experimentos con tres corpus de bases de datos de diferentes dominios.

# CONTENIDO

Resumen	
Contenido.....	i
Lista de Figuras .....	iv
Lista de Tablas .....	v
CAPÍTULO 1. INTRODUCCIÓN	
1.1 Introducción .....	1
1.2 Antecedentes .....	2
1.3 Planteamiento del Problema .....	3
1.4 Descripción de la Complejidad del Problema.....	4
1.5 Limitaciones .....	5
1.6 Objetivos .....	5
1.6.1 Objetivo General .....	5
1.6.2 Objetivos Específicos .....	5
1.7 Justificación .....	6
1.8 Organización de la Tesis .....	6
CAPÍTULO 2. MARCO TEÓRICO	
2.1 Lenguaje .....	8
2.1.1 Lenguaje Natural .....	9
2.1.2 Lenguaje Formal .....	9
2.2 Procesamiento del Lenguaje Natural .....	10
2.3 Bases de Datos .....	10
2.4 Interfaz de Lenguaje Natural .....	11
2.5 Interfaz de Lenguaje Natural a Bases de Datos .....	12
2.6 Interfaces a Bases de Datos .....	13
2.7 Arquitecturas de las ILNBD .....	13
2.8 Structured Query Language (SQL) .....	14

2.9 Álgebra Relacional .....	17
<b>CAPÍTULO 3. ESTADO DEL ARTE</b>	
3.1 Introducción .....	18
3.2 Interfaces Dependientes de Dominio .....	19
3.3 Interfaces Independientes de Dominio .....	19
3.4 Interfaces del Estado del Arte .....	19
3.4.1 LADDER y PLANES .....	19
3.4.2 TQA .....	20
3.4.3 INTELLECT .....	20
3.4.4 PRECISE .....	20
3.4.5 ELF .....	21
3.5 Interfaz del ITCM .....	22
<b>CAPÍTULO 4. PROPUESTA DE SOLUCIÓN</b>	
4.1 Arquitectura de la ILNBD .....	29
4.2 Modulo de Traducción de la ILNBD .....	30
4.3 Tratamiento de la Preposición “de” y de la Conjunción “y” .....	31
4.4 Problemática en el Tratamiento de la Preposición “de” .....	41
4.5 Propuesta de Solución Prevista .....	43
<b>CAPÍTULO 5. IMPLEMENTACIÓN</b>	
5.1 Análisis de la Implementación del Traductor del ITCM .....	48
5.2 Diseño del Tratamiento de la Preposición “de” para más de dos Sustantivos .....	50
5.3 Implementación del Diseño Propuesto para el Tratamiento de la Preposición “de” .....	54
<b>CAPÍTULO 6. RESULTADOS Y CONCLUSIONES</b>	
6.1 Pruebas .....	58
6.2 Resultados Obtenidos para el Corplus de la Base de Datos Northwind .....	61
6.3 Resultados Obtenidos para el Corpus de la Base de Datos Pubs ..	62
6.4 Resultados Obtenidos para el Corpus de la Base de Datos ATIS ..	62

6.5 Conclusiones .....	63
6.6 Trabajos Futuros .....	65
Referencias .....	66

# LISTA DE FIGURAS

Figura 2.1. Arquitectura general de una ILN .....	12
Figura 4.1. Arquitectura de la ILNBD .....	30
Figura 4.2. Fases de la ILNBD .....	31
Figura 4.3. Relación entre el diccionario de Metadatos y el diccionario de Dominio .....	32
Figura 4.4. Porción del diagrama de relaciones de la BD Northwind, donde se observa la relación entre las tablas Products y Categories .....	36

# LISTA DE TABLAS

Tabla 2.1. Estándares de SQL .....	15
Tabla 3.1. ILNBDs del estado del arte .....	22
Tabla 3.2. Resultados obtenidos de la interfaz del Dr. Rojas usando el corpus para la base de datos ATIS .....	27
Tabla 4.1. Casos de sustantivos con la preposición “de” .....	34
Tabla 4.2. Conjuntos de columnas y tablas de los sustantivos Fecha y Nacimiento .....	35
Tabla 4.3. Conjuntos de columnas y tablas de los sustantivos Categoría y Productos .....	35
Tabla 4.4. Resultado de la intersección de los sustantivos Categoría y Productos .....	36
Tabla 4.5. Conjuntos de columnas y tablas de los sustantivos Región y Empleados .....	37
Tabla 4.6. Sustantivos Región y Empleados con valores 1 y 2 respectivamente .....	37
Tabla 4.7. Conjuntos de Columnas y Tablas de los sustantivos Nombre y Empleado .....	38
Tabla 4.8. Columnas y tablas a las que hacen referencia los sustantivos Nombre, Dirección y Teléfono .....	39
Tabla 4.9. Resultado de la unión de los sustantivos Nombre, Dirección y Teléfono .....	40
Tabla 4.10. Columnas a las que hacen referencia los sustantivos Nombre y Contacto .....	41
Tabla 4.11. Columnas que hacen referencia a la intersección entre Nombre y Contacto y al sustantivo Compañía .....	42
Tabla 4.12. Columnas resultantes de la intersección de los sustantivos	

Nombre y Contacto .....	44
Tabla 4.13. Conjunto de columnas resultante de la intersección de los sustantivos Nombre, Contacto, Compañía y Nombre .....	44
Tabla 4.14. Conjunto de columnas resultante de la clausula Where .....	44
Tabla 4.15. Conjuntos resultantes de la clausula Select y Where .....	45
Tabla 4.16. Columnas resultantes de la intersección de los sustantivos Nombre y Ciudad .....	46
Tabla 4.17. Conjunto de columnas resultante de la clausula Where .....	46
Tabla 4.18. Conjunto de columnas y tablas de los sustantivos Nombre, Ciudad y Compañía .....	46
Tabla 5.1. Algoritmos principales de la interfaz .....	49
Tabla 5.2. Algoritmos derivados del Algoritmo General .....	49
Tabla 5.3. Columnas resultantes de la intersección de los sustantivos Nombre y Contacto .....	51
Tabla 5.4. Conjunto de columnas resultante de la intersección de los sustantivos Nombre, Contacto, Compañía y Nombre .....	52
Tabla 5.5. Columnas resultantes de la intersección de los sustantivos Nombre y Ciudad .....	53
Tabla 5.6. Conjunto de Columnas y Tablas de los sustantivos Nombre, Ciudad y Compañía .....	53
Tabla 6.1. Resultados obtenidos con el corpus para la base de datos Northwind con la ILNBD de Rojas .....	61
Tabla 6.2. Resultados obtenidos con el corpus para la base de datos Northwind con la ILNBD de González .....	61
Tabla 6.3. Resultados obtenidos con el corpus para la base de datos Pubs con la ILNBD de Rojas .....	62
Tabla 6.4. Resultados obtenidos con el corpus para la base de datos Pubs con la ILNBD de González .....	62
Tabla 6.5. Resultados obtenidos con el corpus para la base de datos ATIS con la ILNBD de Rojas .....	62

Tabla 6.6. Resultados obtenidos con el corpus para la base de datos ATIS con la ILNBD de González .....	63
--	----

# Capítulo 1

## INTRODUCCIÓN

---

Este capítulo es un preámbulo de la tesis ya que contiene la información básica sobre lo que trata, como la descripción del problema, el objetivo de la investigación y la importancia del proyecto. Además, se mencionan los antecedentes del proyecto y se hace una descripción breve del contenido de cada capítulo.

### 1.1 Introducción

Las bases de datos son el medio de almacenamiento digital más usado. La mayoría de los sitios de Internet, así como negocios o escuelas son solo algunas de las entidades que aprovechan su tecnología. Debido al uso de las bases de datos la consulta a los datos es una actividad realizada cotidianamente y de la cual depende la correcta toma de decisiones en cualquier empresa. Por esta razón, el desarrollo de

sistemas que permitan a los usuarios efectuar esta tarea de forma sencilla, amigable y además eficaz ha sido un área de gran desarrollo en los últimos años.

Para tener acceso a las bases de datos se requiere de conocimientos computacionales, siendo necesario el conocimiento de algún lenguaje de consulta formal, el de más amplio uso es el SQL (Structured Query Language). Debido a que es un lenguaje formal, es difícil para los usuarios inexpertos memorizar las complicadas instrucciones que lo integran, tomando en cuenta que deben conocer la estructura de la base de datos para que puedan realizar las consultas correctamente. Dado este problema se crearon las Interfaces de Lenguaje Natural a Bases de Datos (ILNBDs), a través de las cuales un usuario puede acceder a la información almacenada en una base de datos utilizando una petición en lenguaje natural. El objetivo de las ILNBDs es facilitar a los usuarios que no cuentan con los conocimientos necesarios sobre algún lenguaje formal de bases de datos, el acceso a la información almacenada en una base de datos.

### **1.2 Antecedentes**

Este trabajo sigue una línea de investigación iniciada por Rodríguez en el 2005 [1], en el cual se diseñó una técnica para la traducción de consultas con información implícita a una base de datos. Propone convertir una consulta en lenguaje natural con información implícita al lenguaje formal SQL exitosamente, brindando además independencia de dominio.

Esta técnica de traducción es tomada para el trabajo de Silva en el 2005 [2], en el cuál se aborda el problema de la implementación de un traductor de lenguaje natural independiente del dominio, produciendo una interfaz de este trabajo.

Esta interfaz es tomada para el trabajo de Cruz en el 2006 [3], en el cual se evalúa la interfaz desarrollada por Silva en el 2005 y establece propuestas de solución.

El presente trabajo de tesis, implementando una estrategia de mejora para incrementar el porcentaje de éxito en la interfaz de lenguaje natural desarrollada por Silva en el 2005.

### **1.3 Planteamiento del problema**

En la actualidad existen pocas ILNBDs que utilizan el idioma español, y de ellas, algunas son independientes de dominio, pues éstas requieren configuraciones tediosas para su buen funcionamiento. Esta configuración debe ser realizada para cada dominio que se trate, y esto conlleva una gran carga de trabajo para el administrador del sistema.

En el ITCM se desarrolló una interfaz independiente de dominio en la cual se implementó el tratamiento de la preposición “de”. Se han detectado anomalías tanto en la implementación como en la estructura de la base de datos. Aun cuando esta interfaz logró el objetivo de ser independiente de dominio, requiere incrementar el porcentaje de éxito en las consultas que los usuarios le realizan.

Este trabajo propone una estrategia de mejora para lograr un incremento en el porcentaje de éxito en esta interfaz de lenguaje natural, básicamente enfocado hacia la solución de las problemáticas detectadas en la implementación.

## 1.4 Descripción de la complejidad del problema

NP-Completo es el nombre de una clase de problemas los cuales se sospecha que son particularmente difíciles de procesar. En teoría de la complejidad computacional, la clase de complejidad NP-completo es el subconjunto de los problemas de decisión en NP tal que todo problema en NP se puede reducir en cada uno de los problemas de NP-completo. Se puede decir que los problemas NP-completo son los problemas NP más difíciles y muy probablemente no formen parte de la clase de complejidad P. La razón es que de tenerse una solución polinómica para un problema NP-completo, todos los problemas NP tendrían también una solución en tiempo polinómico (y por lo tanto, si se demuestra que para un problema NP-completo no existe solución en tiempo polinómico, ninguno de los problemas NP tendrá solución).

El procesamiento del lenguaje natural es una tarea difícil debido a la ambigüedad del lenguaje [4]. La complejidad del lenguaje natural es un área importante en el procesamiento del lenguaje, los mecanismos formales pueden ser descritos en términos de complejidad [5].

Un número de investigadores han empleado sus métodos para analizar teorías y modelos lingüísticos, estableciendo una metodología tal como se practica actualmente. De esta forma permiten al programa dar seguimiento y evaluación mediante la lingüística computacional en general. Barton et al. prueba la NP-Complejidad del problema de reconocimiento del lenguaje natural en su libro *Computational Complexity and Natural Language* [6].

## 1.5 Limitaciones

Las limitaciones consideradas para este trabajo son:

1. Los dominios sobre los cuales se realizarán pruebas son ATIS, Northwind y Pubs.
2. Los corpus que se utilizarán serán los utilizados anteriormente por González [38].
3. El traductor no utilizará el módulo de diálogo.

## 1.6 Objetivos

### 1.6.1 Objetivo general

Implementar una estrategia de mejora para incrementar el porcentaje de éxito en una interfaz de lenguaje natural.

### 1.6.2 Objetivos específicos

- Analizar la forma en que los ciclos de una base de datos afectan el proceso de traducción.
- Implementar la creación de vistas para evitar las dificultades de traducción por el diseño de la base de datos.
- Analizar el tratamiento de la preposición “de” implementado en la ILNBD del ITCM [38].

- Adicionar mecanismos a las fases de traducción de consultas de la ILNBD del ITCM, para mejorar su desempeño.

## **1.7 Justificación**

Es muy importante la implementación de una estrategia con el fin de incrementar el porcentaje de éxito en la interfaz de lenguaje natural, que se tomara como base. Debido a que las técnicas de traducción desarrolladas hasta el momento no permiten que esta herramienta tenga un nivel deseable de éxito en las consultas realizadas por los usuarios, considerando las diversas bases de datos que se utilizan para evaluación.

## **1.8 Organización de la tesis**

Capítulo 2. En este capítulo se muestra el soporte teórico de este trabajo de tesis donde se definen los conceptos básicos y necesarios que se involucran en esta investigación.

Capítulo 3. En este capítulo se desarrolla el estado del arte de las ILNBDs, además se detalla específicamente el desarrollo de la interfaz del ITCM que es la base sobre la cual se partió para el presente trabajo de tesis.

Capítulo 4. En este capítulo se describe a detalle el tratamiento de preposiciones y conjunciones en el proceso de traducción de consultas de la ILNBD, se identifica la

problemática existente en este tratamiento y se plantea una propuesta de solución para dicha problemática.

Capítulo 5. En este capítulo se muestra el análisis realizado al código del traductor de la ILNBD del ITCM, destacando los algoritmos principales del traductor que son los que conforman las fases de configuración y de traducción de la interfaz. También se muestra el algoritmo que fue modificado para llevar a cabo la implementación.

Capítulo 6. En este capítulo se muestran las pruebas realizadas con la interfaz una vez teniendo finalizada la etapa de implementación. Se ejemplifican dos tipos de consultas que resuelve esta nueva implementación para el tratamiento de la preposición “*de*”, también se muestran los resultados para los corpus de las bases de datos Northwind, Pubs y ATIS. Finalmente, se presentan las conclusiones a las que se llegó en este trabajo así como también se proponen trabajos futuros y diversas propuestas que ayuden a mejorar la ILNBD para lograr una mayor traducción de consultas.

# Capítulo 2

## MARCO TEÓRICO

---

En este capítulo se muestra el soporte teórico de este trabajo de tesis donde se definen los conceptos básicos y necesarios que se involucran en esta investigación.

### 2.1 Lenguaje

Conjunto de sonidos articulados o símbolos con que el hombre manifiesta lo que piensa o siente. Cuando se habla de lenguajes se pueden diferenciar dos clases muy bien definidas [7] estas son: Los lenguajes naturales (inglés, alemán, español, etc.) y lenguajes formales (matemático, lógico, etc.).

### **2.1.1 Lenguaje natural**

El Lenguaje Natural (LN) es el medio que utilizamos de manera cotidiana para establecer comunicación con las demás personas. Fue desarrollado y organizado a partir de la experiencia humana y puede ser utilizado para analizar situaciones altamente complejas. La riqueza de sus componentes semánticos da a los lenguajes su gran poder expresivo y su valor como una herramienta para el razonamiento.

Se utiliza el término de lenguaje natural para referirse principalmente al lenguaje humano. El término se utiliza en contraposición a los lenguajes formales. Estos son lenguajes artificiales que han sido creados por el hombre con un fin explícito, como por ejemplo, los lenguajes de programación y el lenguaje matemático o lógico [8].

El lenguaje natural evoluciona enmarcado por una cultura de hablantes nativos que utilizan dicho lenguaje con la finalidad de comunicarse. De esta forma, se distinguen como lenguajes naturales los siguientes: el chino mandarín, el español y el inglés.

### **2.1.2 Lenguaje formal**

En matemáticas, lógica, y las ciencias computacionales, un lenguaje formal es un conjunto de palabras de longitud finita formadas a partir de un alfabeto finito.

Informalmente, el término lenguaje formal se utiliza en muchos contextos para referirse a un modo de expresión más cuidadoso y preciso que el habla cotidiana. El lenguaje formal es aquel que el hombre ha desarrollado para expresar las

situaciones específicas en cada área del conocimiento científico. Las palabras y oraciones de un lenguaje formal son perfectamente definidas.

Los lenguajes formales son utilizados para modelar alguna teoría de la mecánica, física, matemática, ingeniería eléctrica, o de otra naturaleza, con la ventaja de que en éstos toda ambigüedad es eliminada [9].

## **2.2 Procesamiento de lenguaje natural**

El procesamiento de lenguaje natural (PLN o NLP en inglés) es un conjunto de técnicas computacionales para analizar y representar textos en uno o más niveles de análisis lingüísticos, con el fin de llevar a cabo el procesamiento del lenguaje humano para un rango de tareas y aplicaciones [10].

## **2.3 Bases de datos**

Una base de datos (BD) es un conjunto de datos relacionados entre sí. Por datos entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito. Una base de datos tiene las siguientes propiedades implícitas:

- Una base de datos representa algún aspecto del mundo real, en ocasiones llamado minimundo o universo de discurso. Las modificaciones del minimundo se reflejan en la base de datos.

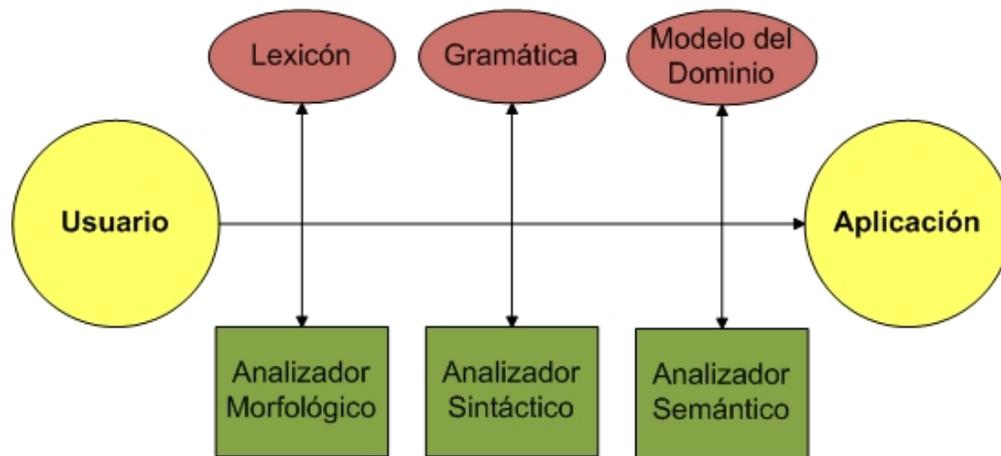
- Una base de datos es un conjunto de datos lógicamente coherente, con cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base de datos.
- Toda base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

Las bases de datos computarizadas se pueden crear y mantener con un grupo de programas de aplicación escritos específicamente para esa tarea, o bien mediante un sistema de gestión de bases de datos.

Un sistema de gestión de bases de datos (SGBD, en inglés, *database management system: DMBS*) es un conjunto de programas que permite a los usuarios crear y mantener una base de datos. Por tanto, el SGBD es un sistema de software de propósito general que facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones [11].

### **2.4 Interfaz de lenguaje natural**

Las Interfaces de Lenguaje Natural (ILN) son mecanismos de comunicación entre una persona y una máquina a través de un lenguaje natural. Por lo general, esta comunicación es bidireccional, es decir, de tipo pregunta-respuesta. La arquitectura general de una ILN se muestra en la Figura 2.1.



**Figura 2.1.** Arquitectura general de una ILN

## 2.5 Interfaz de lenguaje natural a bases de datos

Una interfaz de lenguaje natural a bases de datos es un sistema que permite que un usuario acceda a la información almacenada en una base de datos mediante una solicitud expresada en algún tipo de lenguaje natural [12].

Este sistema funciona como intermediario entre el usuario y la máquina; de tal forma que el usuario puede realizar alguna consulta a una base de datos a través de su idioma natural y la interfaz realizará la traducción a un lenguaje de consultas como el lenguaje SQL. La consulta en SQL es enviada al administrador de bases de datos que la procesa y genera los resultados de ésta y los muestra al usuario como respuesta a su petición hecha en lenguaje natural.

## 2.6 Interfaces a bases de datos

Existe una clasificación de

- Lenguajes Formales de recuperación. El usuario debe conocer algún lenguaje formal como SQL, para poder hacer sus peticiones a la base de datos.
- Interfaces gráficas y menús de selección. El usuario es guiado por menús que indican cómo formular sus consultas.
- Interfaces basadas en formularios. El usuario llena campos para realizar su petición.
- Interfaces de lenguaje natural. El usuario interactúa con la computadora por medio de un lenguaje natural.

## 2.7 Arquitecturas de las ILNBDs

Introducción

- Sistemas de reconocimiento de patrones. Utiliza palabras clave de las oraciones para relacionarlas con patrones establecidos de acuerdo a una base de datos específica.
- Sistemas basados en sintaxis. Están basados en una gramática la cual describe las reglas para la formación de estructuras sintácticas correctas. La oración introducida por el usuario es analizada a nivel sintáctico para la

generación de un árbol de análisis y posteriormente encontrar su correspondencia con la instrucción en lenguaje formal.

- Sistemas de gramática semántica. Se basan en el significado de las palabras dentro de la oración, utilizan una gramática semántica la cual clasifica las relaciones entre las palabras que dan un significado lógico para permitir la comprensión de la petición.
- Lenguajes de representación intermedia. Un intérprete semántico transforma un árbol de análisis sintáctico en un lenguaje intermedio el cual representa el significado de la consulta introducida. Facilita la traducción de la oración a un lenguaje formal.
- Generación de respuestas. Proporciona la respuesta más adecuada al usuario tratando de identificar qué es lo que éste desea obtener.
- Sistemas subyacentes múltiples. Las interfaces que poseen esta arquitectura no solamente recuperan información de una base de datos, también interactúan con otros sistemas que realizan otras funciones como: procesadores de texto, sistemas de correo electrónico, entre otros.

### **2.8 Structured Query Language (SQL)**

SQL (Lenguaje de consultas estructurado) fue desarrollado por IBM, originalmente denominado SEQUEL, como parte del proyecto System R a principios de 1970. Hoy en día numerosos productos son compatibles con el lenguaje SQL, y se ha establecido como el lenguaje estándar para las bases de datos relacionales. La versión más reciente publicada por la ANSI (American National Standards Institute)

es SQL:2008. SQL es una combinación de constructores del álgebra relacional y del cálculo relacional. Usando SQL es posible, además de definir la estructura de los datos, modificar éstos en la base de datos y especificar restricciones de seguridad [13].

En la actualidad el lenguaje SQL es el estándar de uso en la inmensa mayoría de los sistemas de gestión de BD comerciales. La diversidad de añadidos que incluyen las distintas implementaciones comerciales del lenguaje es amplia y el soporte al estándar SQL-92 es general y muy amplio. En la tabla 2.1 se muestran las diferentes revisiones y agregados que a lo largo del tiempo se han realizado a SQL.

**Tabla 2.1.** Estándares de SQL

Año	Nombre	Comentarios
1986	SQL-86	Primera publicación hecha por ANSI. Confirmada por ISO en 1987.
1989	SQL-89	Revisión menor [PROGRESS 2001a].
1992	SQL-92	Revisión mayor [PROGRESS 2001b].
1999	SQL:1999	Se agregaron expresiones regulares, consultas recursivas (para relaciones jerárquicas), disparadores y algunas características orientadas a objetos [EISENBERG 1999].
2003	SQL:2003	Introduce algunas características de XML, cambios en las funciones, estandarización del objeto <i>sequence</i> y de las columnas autonuméricas [EISENBERG 2004].
2006	SQL:2006	Define las maneras en las cuales el lenguaje SQL se puede utilizar conjuntamente con XML. Define formas para importar y guardar datos de XML en una base de datos en SQL, manipulándolos dentro de la base de datos y publicando el XML y los datos en SQL convencionales en forma XML. Además, proporciona facilidades que permiten a las aplicaciones integrar dentro de su código en SQL el uso de XQuery, lenguaje de consulta de XML publicado por el W3C (World Wide Web Consortium) para acceso concurrente a datos ordinarios en SQL y documentos en

		XML [OPEL 2009].
2008	SQL:2008	Permite el uso de la cláusula ORDER BY fuera de las definiciones de los cursores. Incluye los disparadores de tipo INSTEAD OF. Añade la sentencia TRUNCATE [IEEE 2008].

El lenguaje SQL tiene varios componentes [7]:

- *Lenguaje de definición de datos (LDD)*. Proporciona comandos para la definición, borrado y modificación de los esquemas de tablas.
- *Lenguaje de manipulación de datos (LMD)*. Es un lenguaje de consultas basado tanto en el álgebra relacional como en el cálculo relacional de tuplas. También contiene comandos para insertar, borrar y modificar renglones.
- *Integridad*. El LDD incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos. Las actualizaciones que violan las restricciones de integridad se rechazan.
- *Definición de vistas*. El LDD incluye comandos para la definición de vistas.
- *Control de transacciones*. Incluye comandos para especificar el comienzo y fin de las transacciones.
- *SQL intercalado y SQL dinámico*. El SQL intercalado y el SQL dinámico definen cómo se pueden incorporar instrucciones de SQL en lenguajes de programación de propósito general como C, C++, Java, PL/I, Cobol, Pascal y Fortran.
- *Autorización*. El LDD incluye comandos para especificar los derechos de acceso de los usuarios a las relaciones y a las vistas.

## 2.9 Álgebra relacional

En el presente trabajo de investigación se utilizan los términos unión e intersección de conjuntos. Estos términos son operadores del álgebra relacional utilizados en este trabajo para realizar las operaciones correspondientes a los sustantivos de las consultas realizadas por el usuario a la ILNBD. A continuación se definen estos dos términos.

*Unión:* Regresa una relación que contiene todas las tuplas que aparecen en las dos relaciones especificadas.

*Intersección:* Regresa una relación que contiene todas las tuplas que aparecen en las dos relaciones especificadas (en ambas, no en una u otra) [14].

# Capítulo 3

## ESTADO DEL ARTE

---

En este capítulo se desarrolla el estado del arte de las ILNBDs, además se detalla específicamente el desarrollo de la interfaz del ITCM que es la base sobre la cual se partió para el presente trabajo de tesis.

### 3.1 Introducción

Desde los años 60's y hasta la fecha se han desarrollado infinidad de trabajos sobre ILNBDs; sobre todo en los 70s y 80s. A pesar de eso, la investigación en esta rama ha sufrido un estancamiento desde entonces debido a lo complejos que son los problemas a los que se enfrenta el uso de estas interfaces; algunos ejemplos son: la ambigüedad de la interpretación de las palabras, la precisión en las respuestas, la adaptabilidad a nuevos dominios, entre otros [15].

### 3.2 Interfaces dependientes del dominio

La mayoría de las ILNBDs que han logrado obtener buenos resultados han sido aquéllas que son dependientes del dominio. Actualmente estos trabajos se enfocan principalmente a aplicaciones específicas y a algunos idiomas en particular. Algunos de estos trabajos son los descritos en Boonjing 2006 [16], Le 2007 [17], Djahantighi 2008 [18], Ahmad 2009 [19], Kovács 2009 [20], Sarhan 2009 [21] y Rao 2010 [22].

### 3.3 Interfaces independientes del dominio

Existen trabajos que son independientes del dominio, tales como Genlex [23], Gingseng [24], AquaLog, PowerAqua [25, 26], NLP-Reduce [27,28], DaNaLIX [29], Orakel [15], Querix, Panto [30], Lapoon 2008 [31] y FREyA [32]. Éstos tratan de ILNs aplicadas a consultas sobre ontologías, las cuales funcionan como BDs deductivas.

### 3.4 Interfaces del estado del arte

Introducción

#### 3.4.1 LADDER y PLANES

Para finales de la década de los setentas aparecieron interfaces que proponían nuevos diseños basados en el uso de gramáticas. Ladder [33] es una interfaz que puede ser configurada para trabajar con diferentes sistemas manejadores de bases de datos, y la interfaz Planes [34] es capaz de mantener un diálogo con el usuario

con el fin de clarificar la consulta, además de interactuar con una base de datos relacional para vuelos.

### **3.4.2 TQA**

El sistema Transformational Question Answering, TQA, fue usado por un año en el departamento de planeación de una ciudad, para una base de datos que contenía registros de cada parcela de tierra de la ciudad [35]. Se guardaron automáticamente registros de las sesiones, que consistían en una copia de las entradas recibidas en la terminal y una descripción del desarrollo del sistema. De las 788 consultas de entrada, el 65% fueron contestadas. Sin embargo no es posible saber cuántas de esas respuestas fueron verdaderamente útiles para los usuarios.

### **3.4.3 INTELLECT**

La primer ILN comercial, INTELLECT fue evaluada en un estudio de campo de acuerdo a Capindale y Crawford. A 19 usuarios se les dió una introducción al sistema. Después utilizaron libremente la interfaz por un periodo de diez semanas generando sus propias preguntas. Se midió el porcentaje de éxito a través de reportes de las transacciones realizadas.

Otras interfaces independientes de dominio que han obtenido buenos resultados son PRECISE [36] y la interfaz comercial ELF [37].

### **3.4.4 PRECISE**

PRECISE es una interfaz de lenguaje natural que utiliza un algoritmo de flujo máximo y traduce consultas que son semánticamente tratables. Esta interfaz es independiente del dominio, es decir, puede ser configurada para que trabaje con bases de datos de diferentes ámbitos.

Los componentes de esta interfaz son el lexicón, el analizador sintáctico, el tokenizador, el comparador, el verificador de equivalencias y el generador de consultas, a través de los cuales se logra el proceso de traducción.

En el proceso de configuración, el lexicón se encarga de extraer en forma automática los elementos de la base de datos. De forma manual se identifica un conjunto de sinónimos para cada palabra utilizando la ontología de propósito general: Word Net. Cada elemento de la base de datos queda asociado con una palabra, y cada palabra con un conjunto de sinónimos. El lexicón también contiene una serie de restricciones para ciertos tokens (preposiciones y verbos).

PRECISE utiliza un analizador sintáctico estándar (Parser Charniak) con el que se realiza el análisis y después se extraen relaciones de enlace entre los tokens. El tokenizador busca en el lexicón los tokens en los que se encuentra la raíz de cada palabra y recupera el conjunto de elementos de la base de datos correspondientes para cada token. El Comparador construye un grafo atributo-valor, ejecuta el algoritmo de Máximo Flujo y verifica si se satisfacen las restricciones sintácticas representadas por los tokens ligados.

Posteriormente se prueba si éstas pueden ser traducidas en consultas SQL. Si se han determinado diferentes interpretaciones, PRECISE solicita al usuario que elija alguna de ellas para dar su respuesta. El proceso final consiste en generar la sentencia SQL correspondiente a la consulta. Si contiene múltiples relaciones se agregan condiciones de unión a la cláusula WHERE.

### **3.4.5 ELF**

ELF es un sistema comercial que genera Sistemas de Procesamiento de Lenguaje Natural para una base de datos. Fue desarrollado por ELF Software Co. ELF es una interfaz que trabaja con Microsoft Access y Visual Basic.

Tabla 3.1. ILNBDs del estado del arte

INTERFAZ	Dependientes del dominio				Independientes del dominio		
MÉTODO	LADDER	PLANES	TQA	INTELLECT	Precise	ELF Software	ITCM
Entrenamiento	✓	✓	✓	✓			✓
al Usuario							
Elaboración					✓	✓	✓
de Corpus							
Evaluación del	✓		✓	✓	✓	✓	✓
porcentaje de éxito							
Configuración					Automática	Automática	Automática
Lenguaje Formal	SQL	Lenguaje Formal	SQL	SQL	SQL	SQL	SQL

### 3.5 Interfaz del ITCM

La interfaz del ITCM fue producto de un trabajo de tesis doctoral desarrollado por González [38]. Esta tesis generó diversos trabajos, los cuales aportaron una parte a la interfaz. Algunos otros trabajos también derivados de esta tesis, no fueron tomados como aportación directa a la interfaz, sin embargo, lo fueron para mostrar porqué éstos no era adecuado tomar en cuenta y así tener bases para saber cuáles trabajos ofrecen una aportación positiva ayudando a cumplir con los objetivos de la interfaz. A continuación se describen los trabajos de tesis derivados.

“Implementación de un Analizador Gramatical del Lenguaje Español”, tesis de maestría desarrollada por la alumna Ana Patricia Domínguez Sánchez, Agosto de

2002 [39]. En este trabajo de investigación se realizó una revisión de los proyectos de interfaces de lenguaje natural para consultas a bases de datos y las técnicas de traducción utilizadas y se implementó un analizador gramatical para ser usado en una ILNBD de dominio específico. Con la información obtenida se pudo determinar la importancia de la independencia del dominio de una ILNBD y la dificultad que ésta implica. Esta tesis aportó el estado del arte para el proyecto doctoral.

“Diseño de la Interfaz de Lenguaje Natural para Consultas a Bases de Datos”, tesis de licenciatura realizada por la alumna Erika Alarcón Ruiz, Agosto de 2002 [40]. En este trabajo se presentó el diseño y la implementación de una interfaz de lenguaje natural para consultar la base de datos de alumnos del ITCM vía internet; realiza la interpretación de una consulta en lenguaje natural utilizando el analizador gramatical desarrollado por Ana Patricia Domínguez. Este proyecto permitió desarrollar una ILNBD para ser usada en Internet, pero no aportó ningún elemento a la tesis doctoral.

“Desarrollo de un Módulo de Predicados Lógicos para Consultas a la Base de Datos de Alumnos del ITCM”, tesis de licenciatura desarrollada por el alumno Alejandro Mendoza Mejía, Agosto de 2002 [41]. En este proyecto se desarrolló el analizador semántico para la ILNBD implementada por Erika Alarcón Ruíz utilizando predicados lógicos. Este analizador está diseñado para determinar el significado de las consultas más comunes a la base de datos de alumnos del ITCM.

Con las tesis de licenciatura de Erika Alarcón Ruíz y de Alejandro Mendoza Mejía, se desarrolló una ILNBD dependiente del dominio. La aportación de ambos proyectos de licenciatura al proyecto doctoral, fue clarificar las desventajas del uso de la técnica de predicados lógicos y mostrar la dificultad para lograr la independencia del dominio.

“Consultas a una Base de Datos en Lenguaje Natural Utilizando Múltiples Tablas”, tesis de licenciatura realizada por las alumnas María del Rosario Smith Salas y Karen Castellanos Jongitud, Diciembre de 2003 [42]. Este proyecto es una continuación de los trabajos de licenciatura anteriores, en el cual se desarrolló una ILNBD utilizando la técnica de predicados lógicos para acceder a la información de una sola tabla. Además, se desarrollo una técnica de predicados lógicos que involucran el uso de múltiples tablas.

“Diseño de una Técnica Basada en Grafos Semánticos para la Traducción de Consultas de Lenguaje Natural a Lenguaje Formal”, tesis de maestría realizada por la alumna Erika Alarcón, Agosto de 2004 [43]. Este trabajo presenta una arquitectura para la traducción de consultas en español a SQL sin la necesidad de tediosas configuraciones. La independencia del dominio fue el principal problema a resolver. En este proyecto se diseñó una técnica de traducción para consultas explícitas, utilizando un grafo semántico que organiza la información de cada palabra de la consulta según el contexto de la base de datos que se está utilizando. El diseño muestra el tratamiento de la preposición “de” y la conjunción “y” utilizando la teoría de conjuntos, lo cual ayuda al análisis semántico del traductor. En esta tesis quedó plasmado parte del diseño de la propuesta doctoral.

“Construcción de un Preprocesador de Consultas en Lenguaje Natural a una Base de Datos”, tesis de maestría desarrollada por el alumno Alejandro Mendoza Mejía, Noviembre de 2004 [44]. En este trabajo se implementó un preprocesador de consultas cuyo objetivo es construir un diccionario de dominio de una manera fácil, para lo cual fue necesario construir un diccionario de sinónimos general para que pudiera ser utilizado por cualquier base de datos y un diccionario de metadatos que es generado de manera automática de acuerdo con la base de datos que se está utilizando. El preprocesador también analiza la consulta formulada por el usuario y la etiqueta con información léxica, sintáctica y semántica. El proyecto doctoral utiliza la arquitectura y el código del preprocesador desarrollado por esta tesis.

“Representación de una Consulta de Lenguaje Natural a través de un Grafo”, tesis de licenciatura realizada por el alumno José Francisco Delgado Orta, Marzo de 2005 [45]. En esta tesis se realiza la implementación de un grafo semántico ponderado para representar una consulta explícita. Algunas funciones de código para la construcción del grafo, son utilizadas en el proyecto doctoral.

“Construcción de un Diccionario de Valores para una Interfaz de Lenguaje Natural para Bases de Datos”, proyecto de titulación para licenciatura opción IV, realizado por la alumna Yuridia Torres Romero, Mayo de 2005 [46]. En este proyecto se construyó un diccionario que almacena los valores de tipo cadena que se encuentran en una base de datos. Este diccionario es útil cuando la información almacenada en la base de datos no se actualiza constantemente; sin embargo, este diccionario no fue utilizado en el proyecto doctoral.

“Diseño de una Técnica para la Traducción de Consultas con Información Implícita a una Base de Datos”, tesis de maestría realizada por la alumna Myriam Janeth Rodríguez Martínez, Octubre de 2005 [1]. Esta tesis es resultado de la tipificación de consultas obtenida del corpus. La relación de las consultas implícitas no era parte de los objetivos de la tesis doctoral; sin embargo, dada la importancia y el avance en el diseño de la técnica para resolver este tipo de consultas, se agregó al proyecto doctoral. El diseño desarrollado en este trabajo modela el significado de una consulta en lenguaje natural mediante un grafo semántico de la base de datos. En esta tesis se utiliza el diccionario de valores desarrollado por Yuridia Torres Romero, debido a que los valores tipo cadena aportan información importante en el proceso de traducción.

“Implementación de un Traductor de Lenguaje Natural Independiente del Dominio”, Tesis de maestría realizada por el alumno Omar Ulises Silva Domínguez, Diciembre de 2005 [2]. En esta tesis se implementó un traductor de consultas en lenguaje

natural en español a SQL independiente de dominio. La arquitectura propuesta consta de dos grandes módulos, el preprocesador y la técnica de traducción, de esta manera se separan los datos que son dependientes del dominio de los datos que pueden ser tratados de manera general, logrando así la independencia de dominio. El traductor es pionero en el uso de la teoría para el tratamiento de la preposición “de” y conjunción “y”, realizando una intersección de conjuntos para las preposiciones y una unión de conjuntos para las conjunciones. También puede identificar a qué se hace referencia, con valores que no se especifican explícitamente, cuando éstos se encuentran en la parte de la oración que corresponde a la cláusula WHERE.

“Evaluación de una Interfaz de Lenguaje Natural a Bases de Datos en Idioma Español”, Tesis de maestría realizada por la alumna Irma Cristina Cruz Castro, Noviembre de 2006 [3]. En este trabajo se realizó un entrenamiento a los usuarios, se creó un corpus de consultas, se realizó una evaluación de consultas y se analizaron los resultados, todo esto con el propósito de evaluar la interfaz con un dominio distinto a los evaluados en otros trabajos de tesis anteriores y así determinar la funcionalidad de la misma. Además, al realizar esta evaluación se encontraron áreas de oportunidad para poder mejorar el funcionamiento de la interfaz.

“Administrador de Diálogo para una Interfaz de Lenguaje Natural a Bases de Datos”, Tesis doctoral realizada por el MCC. Juan Carlos Rojas Pérez, Junio de 2009 [7]. Este trabajo constituye una continuación de la ILNBD desarrollada en el CENIDET, al traductor de ésta se le incorporó un administrador de diálogo. Este administrador de diálogo está conformado por procesos de diálogo que fueron diseñados a partir de una tipificación de problemas en consultas, la cual fue desarrollada del análisis de tres corpus de consultas para las bases de datos Northwind, Pubs y ATIS. La tipificación de consultas clasifica éstas de acuerdo a varios tipos de problemas en consultas, relacionados principalmente al problema de “economía de palabras”. Esta tipificación posee dos características importantes: generalidad, lo que permite su

implementación en varios lenguajes como: inglés, francés, italiano, etc.; e independencia de dominio, es decir, puede ser aplicada a cualquier base de datos relacional. Los procesos de diálogo implementados a partir de la tipificación heredan las dos características, además de que conducen al usuario a obtener información que está buscando.

Los trabajos mencionados anteriormente conforman una parte del desarrollo de la INLBD, ésta también fue tomada como base de algunos trabajos desarrollados en el Instituto Tecnológico de León, una prueba de esto es el trabajo titulado “Memoria para una Interfaz de Lenguaje Natural a Base de Datos”, artículo publicado en el CIIC’10 International Congress on Computer que se llevo a cabo en el Instituto Tecnológico de Ciudad Madero, publicado por la MC. María Dolores Juárez R., Octubre de 2010 [47].

En el presente trabajo de tesis se propone una estrategia de mejora para incrementar el porcentaje de éxito en la traducción de consultas de lenguaje natural. Teniendo dos versiones distintas de la ILNBD originalmente desarrollada en el CENIDET se procedió a evaluar ambas interfaces para determinar cuál de éstas sería la base para esta nueva aportación. Se probó el corpus obtenido del trabajo de tesis de [3] en la base de datos ATIS con la interfaz con diálogo, para compararla con la interfaz desarrollada en el CENIDET en base a los resultados ya obtenidos en el trabajo de Cruz. Los resultados obtenidos en esta fase de evaluación se muestran en la tabla 3.2.

**Tabla 3.2.** Resultados obtenidos de la interfaz (última versión) del Dr. Rojas usando el corpus para la base de datos ATIS.

<b>INTERFAZ</b>	<b>ATIS</b>	<b>CORRECTAS</b>		<b>INCORRECTAS</b>	
Dr. González	Consultas	43	61.4%	27	38.6%
Dr. Rojas	Consultas	23	32%	47	67%
	Sin Traducción	0		12	
	Traducidas	23		35	

En base a los resultados obtenidos en esta fase de evaluación de interfaces se determinó tomar como base la interfaz desarrollada en el CENIDET por el Dr. González, para realizar la implementación del presente trabajo.

# Capítulo 4

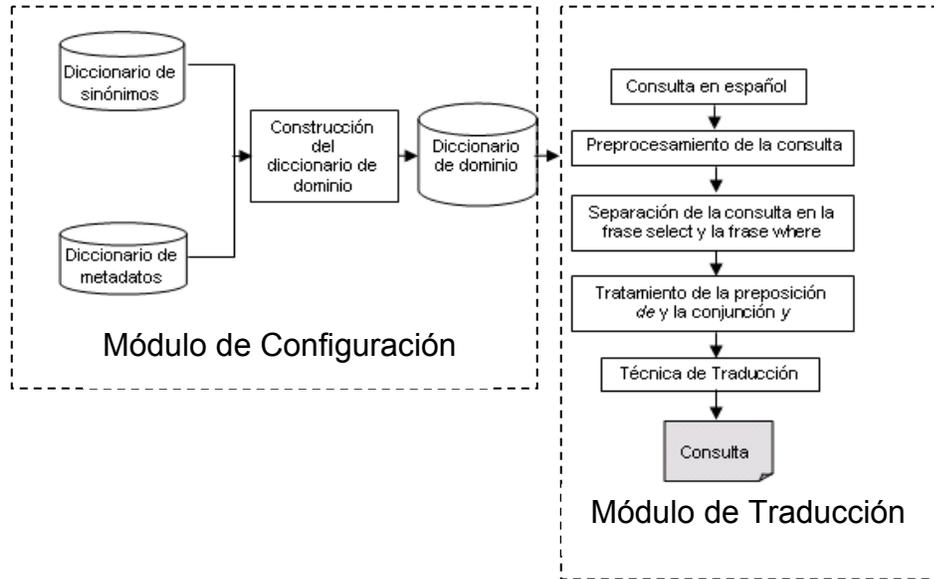
## PROPUESTA DE SOLUCIÓN

---

En este capítulo se describe a detalle el tratamiento de preposiciones y conjunciones en el proceso de traducción de consultas de la ILNBD, se identifica la problemática existente en este tratamiento y se plantea una propuesta de solución para dicha problemática.

### 4.1 Arquitectura de la ILNBD

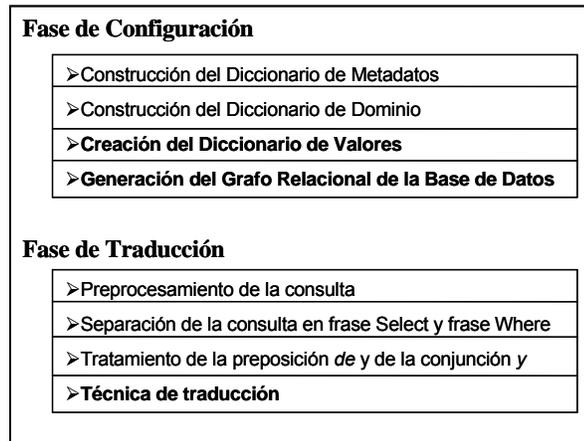
La arquitectura en la que se empleó el tratamiento de la preposición “de” se presenta en el esquema de la figura 4.1, el cual representa una ILNBD independiente de dominio [2].



**Figura 4.1.** Arquitectura de la ILNBD

## 4.2 Modulo de traducción de la ILNBD

Dentro del esquema de arquitectura de la ILNBD se encuentran los módulos de configuración y traducción de la interfaz. El primer módulo referente a la configuración es el módulo que hace posible la independencia de dominio de la interfaz, el segundo módulo, llamado módulo de traducción es donde se lleva a cabo la traducción de la consulta de lenguaje natural a lenguaje formal. En la figura 4.2 se muestran las fases de la ILNBD.



**Figura 4.2.** Fases de la ILNBD

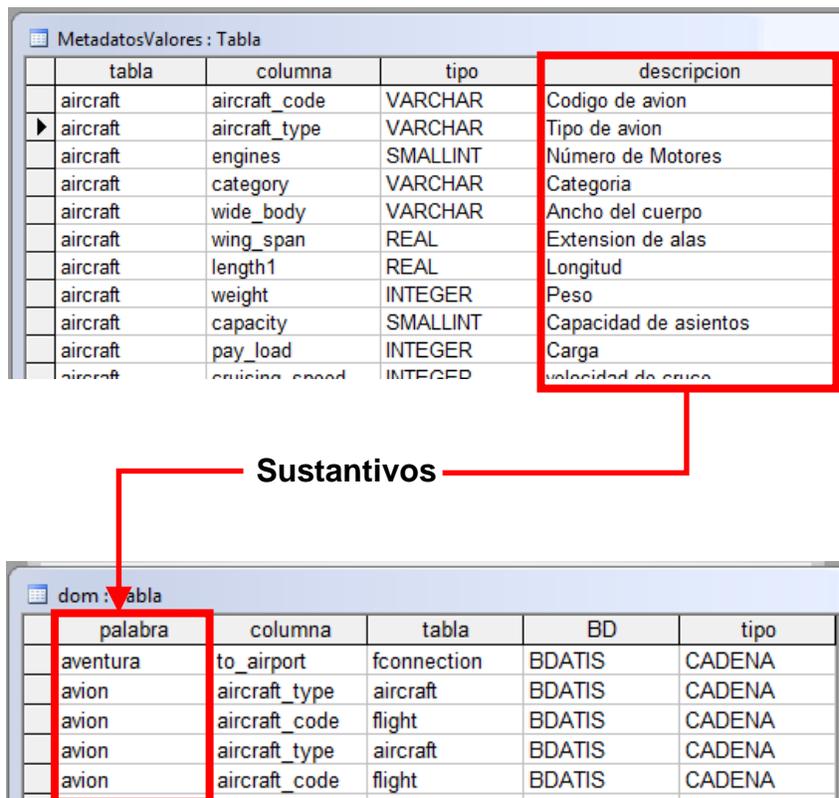
### 4.3 Tratamiento de la preposición “*de*” y de la conjunción “*y*”

En el tratamiento de la preposición “*de*” y la conjunción “*y*” se le aplica un tratamiento especial a la consulta del usuario, tomando sólo la parte que se detectó en la fase anterior como parte de la frase *Select*. Los elementos que toman un papel importante en esta fase son los sustantivos, la preposición “*de*” y la conjunción “*y*”, las comas también son consideradas como conjunción “*y*”, estos son los únicos elementos tomados en cuenta para esta fase.

Para el tratamiento de las preposiciones y las conjunciones se utiliza la teoría de conjuntos. Debido a que el lenguaje natural es muy variado, en especial, el idioma español, el representarlo completamente es una tarea difícil; por esta razón el traductor se dedica a trabajar con un subconjunto del idioma, el cual es llamado dominio, cada base de datos representa un dominio específico. El manejo de las

preposiciones y las conjunciones es tratado mediante operaciones de intersección y unión de conjuntos respectivamente.

La técnica de traducción del traductor utiliza los sustantivos encontrados en la columna descripciones de la tabla de metadatos, en donde cada sustantivo hace referencia a un conjunto de columnas, como se puede observar en la tabla del diccionario de dominio, que se muestra en la figura 4.3.



**Figura 4.3.** Relación entre el diccionario de Metadatos y el diccionario de Dominio

El tratamiento aplica a los sustantivos encontrados en la consulta del usuario, y para cada sustantivo se extrae del diccionario de dominio aquellas columnas a las cuales hace referencia. La manera en la que se ligan esos sustantivos es mediante las preposiciones y conjunciones, los cuales aportan información relevante al traductor.

Cada sustantivo de la oración del usuario hace referencia a un conjunto de columnas, esto también se puede observar en la Figura 4.3, pero no se sabe la relación que puede haber entre los sustantivos, ahí es donde entran en función las preposiciones y conjunciones, que de acuerdo a su significado en el idioma español se asemejan a las operaciones de intersección y unión de conjuntos.

El proceso para el tratamiento de las preposiciones y conjunciones es el siguiente:

1. Se obtiene la parte de la oración del usuario que corresponde a la parte *Select* detectada en la fase 1. Esta oración sólo contendrá sustantivos, preposiciones y conjunciones, las demás palabras son ignoradas.
2. A cada sustantivo le acompaña su valor numérico, que como se comentó anteriormente, indica a qué elementos hace referencia el sustantivo, 0 si no hace referencia ni a columnas ni a tablas, 1 si hace referencia a columnas y 2 si hace referencia a tablas. A las preposiciones y conjunciones se les asigna el valor numérico de -1.
3. Para cada sustantivo en la oración se obtiene un conjunto de columnas y tablas a las que hace referencia, esta información se encuentra en el diccionario de dominio.
4. La preposición “*de*” actúa como una intersección de conjuntos entre los sustantivos o conjuntos resultantes de operaciones previas. La conjunción “*y*” actúa como una unión de conjuntos entre los sustantivos o conjuntos resultantes de operaciones previas.

**Análisis de la preposición “*de*”.** La preposición “*de*” en el idioma español establece una estrecha relación entre los sustantivos previos y posteriores a ésta, también indica pertenencia; tomando en cuenta lo anterior se deduce que cuando los sustantivos previo y posterior tienen el mismo valor numérico se establece una relación entre ellos, y cuando el sustantivo previo tiene el valor numérico de 1 y el posterior tiene valor numérico de 2, indica pertenencia de una columna en una tabla.

Si ambos sustantivos tienen valor numérico de 1 cuando se lleva a cabo la intersección el resultado será el conjunto de columnas en las que ambos sustantivos aparecen juntos en una descripción de columna. Cuando ambos sustantivos tienen valor de 2 y existe intersección siempre y cuando exista una relación directa entre esas tablas, el conjunto obtenido estará conformado por las columnas y tablas a las que hacen referencia ambos sustantivos. La tabla 4.1 muestra los casos que se manejan entre los sustantivos y la preposición “de”.

**Tabla 4.1.** Casos de sustantivos con la preposición “de”

Caso	Sustantivo	Preposición	Sustantivo	Resultado
1. Relación	1	de	1	Ambos sustantivos deben aparecer juntos en la descripción de una columna.
2. Relación	2	de	2	Columnas y Tablas a las que hace referencia ambos sustantivos si y solo si existe relación directa entre las tablas.
3. Pertenencia	1	de	2	Pertenencia de una columna en una tabla.

**Caso 1:** Ambos sustantivos tienen el valor numérico de 1, en la tabla 4.2 se muestra el conjunto de columnas a las que hace referencia cada sustantivo, del fragmento de oración “*fecha de nacimiento*”. Este ejemplo está basado en la base de datos Northwind.

**Tabla 4.2.** Conjuntos de columnas y tablas de los sustantivos *fecha* y *nacimiento*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<b>fecha</b>	Employees.BirthDate Employees.HireDate Orders.OrderDate Orders.RequiredDate Orders.ShippedDate	∅
<b>nacimiento</b>	Employees.BirthDate	∅

Al realizar la intersección de conjuntos sólo se toman en cuenta aquellas columnas que son comunes para ambos sustantivos. Es decir, aquellas columnas que tengan en su descripción ambos sustantivos. En la tabla 4.2 se observan remarcadas las columnas *Employees.BirthDate* de ambos conjuntos.

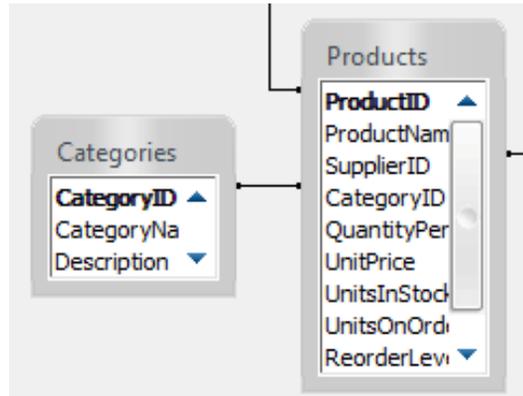
**Caso 2:** Cuando ambos sustantivos tienen un valor numérico de 2, tenemos este ejemplo: “*categorías de productos*”; en la tabla 4.3 se muestran las columnas y tablas a las que hacen referencia cada sustantivo.

**Tabla 4.3.** Conjuntos de columnas y tablas de los sustantivos *categorías* y *productos*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<b>categorías</b>	Categories.CategoryID Categories.CategoryName Products.CategoryID	Categories
<b>productos</b>	OrderDetails.ProductID Products.ProductID Products.ProductName	Products

Cuando ambos sustantivos tienen valor numérico de 2 y se realiza la intersección, el conjunto obtenido estará conformado por las columnas y tablas a las que hacen

referencia ambos sustantivos, siempre y cuando exista una relación directa entre esas tablas, la figura 4.4 muestra la relación que existe entre las tablas *Categories* y *Products*.



**Figura 4.4.** Porción del esquema de la BD Northwind, donde se observa la relación entre las tablas *Products* y *Categories*

El resultado de la intersección se muestra en la tabla 4.4 entre los conjuntos de tablas y columnas de los sustantivos *categorías* y *productos*:

**Tabla 4.4.** Resultado de la intersección de los sustantivos *Categoría* y *Producto*

Conjunto de Columnas Resultante	Conjunto de Tablas Resultante
Products.CategoryID	Categories Products

**Caso 2 extendido:** Cuando ambos sustantivos tengan el valor numérico de 2, puede ser que el sustantivo previo haga referencia a tablas, pero además, hace referencia a columnas; es de esperar que el usuario quiera hacer referencia a una columna y no a una tabla. Por ejemplo, para la oración: *“región de empleados”* tomando en cuenta que ambos sustantivos tienen valor numérico 2, la tabla 4.5 muestra las tablas y columnas a las que hace referencia cada sustantivo.

**Tabla 4.5.** Conjuntos de columnas y tablas de los sustantivos *región* y *empleados*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<b>región</b>	Customers.Region Employees.Region Orders.ShipRegion Region.RegionID Region.RegionDescription Suppliers.Region Territories.RegionID	Region
<b>empleados</b>	Employees.EmployeeID Employees.LastName Employees.FirstName EmployeesTerritories.EmployeeID Orders.EmployeeID	Employees EmployeeTerritories

Para dar resultado a este problema cuando se encuentre el caso en que ambos sustantivos tienen un valor numérico de 2, se verificará primero si lo que quiere el usuario es una columna de una tabla; para lograr ésto, se analizan las columnas a las que hace referencia el sustantivo previo y si al menos una de las columnas que no sea llave foránea pertenece a una tabla del conjunto de tablas del sustantivo posterior, significa que el usuario requiere de una columna de una tabla y es necesario cambiar el valor numérico de 2 del sustantivo previo a un valor de 1 para poder tratarlo como una columna. En la tabla 4.6 se muestra que el sustantivo “*región*” no referencia a ninguna tabla, ya que se ha cambiado el valor a 1.

**Tabla 4.6.** Sustantivos *región* y *empleados* con valores 1 y 2 respectivamente

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<b>región</b>	Customers.Region <b>Employees.Region</b> Orders.ShipRegion Region.RegionID Region.RegionDescription Suppliers.Region Territories.RegionID	∅
<b>empleados</b>	Employees.EmployeeID Employees.LastName	<b>Employees</b> <b>EmployeeTerritories</b>

	Employees.FirstName EmployeesTerritories.EmployeeID Orders.EmployeeID	
--	---	--

Después de cambiar los valores de los sustantivos se procede a realizar la intersección de conjuntos de las columnas, de acuerdo al caso 3 donde el resultado se ve resaltado en la tabla 4.6.

**Caso 3:** Cuando el sustantivo previo tiene un valor de 1 y el posterior un valor de 2, indica una pertenencia de una columna a una tabla. Por ejemplo para la siguiente oración: “*nombre de empleados*”, la tabla 4.7 muestra las tablas y columnas a las que hace referencia cada sustantivo.

**Tabla 4.7.** Conjunto de Columnas y Tablas de los sustantivos *nombre* y *empleados*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<b>nombre</b>	Categories.CategoryName Customers.CompanyName Customers.ContactName Employees.LastName Employees.FirstName Orders.ShipName Products.ProductName Shippers.CompanyName Suppliers.CompanyName Suppliers.ContactName	∅
<b>empleados</b>	Employees.EmployeeID Employees.LastName Employees.FirstName EmployeesTerritories.EmployeeID Orders.EmployeeID	Employees EmployeeTerritories

El resultado serán aquellas tablas a las que hace referencia el sustantivo posterior que tengan alguna de las columnas a las que hace referencia el sustantivo previo, como se muestra en la tabla 4.7.

**Análisis de la conjunción “y”.** La conjunción “y” en el idioma español expresa una idea de suma o acumulación, es por eso que es más sencillo este análisis que el de la preposición “de”. El traductor realiza la unión de conjuntos sólo cuando ambos sustantivos (previo y posterior) tienen el mismo valor numérico.

Un ejemplo sería la oración: “*nombres, direcciones y teléfonos*”, todos los sustantivos tienen valor numérico de 1, las columnas a las que hacen referencia los sustantivos se muestran en la tabla 4.8.

**Tabla 4.8.** Columnas y tablas a las que hacen referencia los sustantivos *nombres, direcciones y teléfonos*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<b>nombres</b>	Categories.CategoryName Customers.CompanyName Customers.ContactName Employees.LastName Employees.FirstName Orders.ShipName Products.ProductName Shippers.CompanyName Suppliers.CompanyName Suppliers.ContactName	∅
<b>direcciones</b>	Customers.Address Employees.Address Employees.PhotoPath Orders.ShipAddress Suppliers.Address	∅
<b>teléfonos</b>	Customers.Phone Employees.HomePhone Shippers.Phone Suppliers.Phone	∅

El resultado de la unión de los conjuntos de columnas de estos sustantivos quedaría como se observa en la tabla 4.9.

**Tabla 4.9.** Resultado de la unión de los sustantivos nombres, direcciones y teléfonos

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<p><b>Unión de Sustantivos</b>  nombres  direcciones  teléfonos</p>	Categories.CategoryName Customers.CompanyName Customers.ContactName Employees.LastName Employees.FirstName Orders.ShipName Products.ProductName Shippers.CompanyName Suppliers.CompanyName Suppliers.ContactName Customers.Address Employees.Address Employees.PhotoPath Orders.ShipAddress Suppliers.Address Customers.Phone Employees.HomePhone Shippers.Phone Suppliers.Phone	<p style="text-align: center;">∅</p>

Un comportamiento similar ocurre cuando se realiza una unión entre sustantivos con valor numérico de 2.

#### 4.4 Problemática en el tratamiento de la preposición “de”

Las reglas establecidas hasta ahora para el tratamiento han logrado obtener resultados favorables para la traducción. Sin embargo, existen algunos casos que no están actualmente contemplados dentro del tratamiento. Por ejemplo:

*Dame el nombre del contacto de la compañía de nombre “Alfredo Futterkiste”*

Para esta consulta el tratamiento de preposiciones realiza las siguientes operaciones:

$$nombre \cap contacto \cap compañía \cap nombre$$

La primera intersección que se realiza es entre los sustantivos *nombre* y *contacto* lo cuales son un caso tipo 1 para el tratamiento de la preposición del cual significa que ambos sustantivos se refieren a columnas, por lo tanto, el resultado deben ser aquellas columnas que tengan en común ambos sustantivos. Esto se muestra en la tabla 4.10.

**Tabla 4.10.** Columnas a las que hacen referencia los sustantivos *nombre* y *contacto*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
nombre	Categories.CategoryName	∅
	Customers.CompanyName	
	Customers.ContactName	
	Employees.LastName	
	Employes.FirstName	
	Orders.ShipName	
	Products.ProductName	
	Shippers.CompanyName	
	Suppliers.CompanyName	
	Suppliers.ContactName	
contacto	Customers.ContactName	∅
	Customers.ContactTitle	

	Suppliers.ContactName Suppliers.ContactTitle	
--	---	--

Posteriormente se interceptan las columnas del resultado anterior con las columnas del sustantivo *compañía*, el cual tiene un valor de 1, ya que hace referencia a columnas. En la tabla 4.11 se pueden observar las columnas a las que hace referencia el sustantivo *compañía*.

**Tabla 4.11.** Columnas que hacen referencia a la intersección entre *nombre* y *contacto* y al sustantivo *compañía*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
Resultado de Sustantivos nombre contacto	Customers.ContactName Suppliers.ContactName	∅
compañía	Customers.CompanyName Shippers.CompanyName Suppliers.CompanyName	∅

El resultado de la intersección de estos dos conjuntos de columnas es vacío, ya que no tienen elementos en común ambos conjuntos y al realizar la siguiente intersección con el sustantivo *nombre*, obtendríamos también un conjunto vacío, debido a la intersección del conjunto vacío resultante de la intersección de la tabla 4.11 con el conjunto de atributos del sustantivo *nombre*. Al obtener como resultado final un conjunto vacío, provoca que no haya una traducción de la consulta, lo cual no ayuda a mejorar el desempeño de la interfaz.

Esta situación se presenta con este tipo de consultas, en las que por medio de la preposición “de” se unen diferentes sustantivos, incluyendo los que indican a qué entidad pertenece la información que se está solicitando: por ejemplo. “*nombre del contacto de la compañía de nombre*”.

Por esta razón, se propone una estrategia que mejore el tratamiento de preposiciones existente para contemplar este tipo de consultas.

#### 4.5 Propuesta de solución preliminar

Una posible solución a esta problemática consiste en tomar ventaja de las características de la consulta. Si la consulta incluye una condición y el valor presente es de tipo cadena, de acuerdo al proceso de traducción, se lograría identificar la columna a la que pertenece, por medio de la búsqueda del valor. Si se cuenta con este valor, entonces se puede tener información acerca de las columnas requeridas en la frase *Select*.

Analizando el proceso para el tratamiento de preposiciones, se determinó que en algunos casos, cuando hay más de dos sustantivos pudiera ser que entre ellos la intersección arroje resultados, pero con un tercer o un cuarto sustantivo ya no exista relación y el conjunto final que se obtiene es vacío. Otro caso es cuando no se obtienen columnas en común con los sustantivos.

Para la interfaz no es posible saber previamente entre cuáles sustantivos habrá relación. Para no alterar las reglas existentes para el tratamiento, se plantea realizar un proceso que determine cuándo se requiere utilizar la información de la frase *Where*, como se muestra a continuación.

##### **Ejemplo1:**

*Dame en nombre del contacto de la compañía de nombre "Alfredo Futterkiste"*

1. Se realizan normalmente las operaciones de intersección o unión, se guardan los conjuntos resultantes que tengan columnas.

**Tabla 4.12.** Columnas resultantes de la intersección de los sustantivos *nombre* y *contacto*

Intersección	Resultado
nombre $\cap$ contacto	Customers.ContactName Suppliers.ContactName

La tabla 4.12 muestra el resultado de esta operación, donde se observa que la intersección entre los primeros 2 sustantivos es diferente de vacío, sin embargo al seguir realizando las operaciones de intersección con el resto de los sustantivos se obtiene finalmente un conjunto vacío como se muestra en la tabla 4.13.

**Tabla 4.13.** Conjunto de columnas resultante de la intersección de los sustantivos *nombre*, *contacto*, *compañía* y *nombre*

Intersección	Resultado
nombre $\cap$ contacto $\cap$ compañía $\cap$ nombre	$\emptyset$

2. Se realiza la fase siguiente en el proceso de traducción que consiste en determinar a qué columna se hace referencia en la condición.

**Tabla 4.14.** Conjunto de columnas resultante de la frase *Where*

Condición	Columnas
Alfredo Futterkiste	Customers.CompanyName Orders.ShipName

Si el resultado de la frase *Select* fue vacío, entonces se utilizará la información de la frase *Where* para tratar de identificar la columna probable solicitada. La tabla 4.14 muestra las columnas resultantes de la frase *Where*.

3. Se utilizan los conjuntos que contienen columnas, si no existen, se utilizarán los conjuntos de columnas correspondientes a los sustantivos involucrados en el tratamiento de la preposición “*de*”.

Aquellas columnas de los conjuntos cuya tabla sea igual a la tabla de la columna o columnas encontradas en la parte de la consulta que corresponde a la frase *Where*. Esto se puede observar en la tabla 4.15, donde se presentan los conjuntos resultantes de las frases *Select* y *Where*.

**Tabla 4.15.** Conjuntos resultantes de la clausula *Select* y *Where*

Columnas de la clausula <i>Select</i>	Columnas de la clausula <i>Where</i>
Customers.ContactName	Customers.CompanyName
Suppliers.ContactName	Orders.ShipName

Con esta operación se habrá identificado la columna solicitada. A diferencia de lo que ocurría con el tratamiento anterior, el cual producía un conjunto vacío.

Otro ejemplo que muestra esta problemática es el siguiente:

**Ejemplo 2:**

*Dame el nombre de la ciudad de la compañía “Exotic Liquids”*

1. Se realizan las operaciones de intersección o unión, se guardan los conjuntos resultantes si se obtuvieron columnas.

**Tabla 4.16.** Columnas resultantes de la intersección de los sustantivos *nombre* y *ciudad*

Intersección	Resultado
nombre $\cap$ ciudad	$\emptyset$

En la tabla 4.16 se puede observar que al hacer la intersección entre los primeros sustantivos se obtiene un conjunto vacío como resultado, lo cual conlleva a obtener como resultado final un conjunto vacío al interceptarse este conjunto resultante con el sustantivo *compañía*.

2. Se realiza la fase siguiente en el proceso de traducción para determinar a qué columna se hace referencia en la condición. La tabla 4.17 muestra el conjunto de columnas resultante.

**Tabla 4.17.** Conjunto de columnas resultante de la frase *Where*

Condición	Columnas
Exotic Liquids	Suppliers.CompanyName

3. Se utilizan los conjuntos donde se obtuvieron columnas, en este caso no existen conjuntos resultantes de alguna intersección, entonces se utilizarán los conjuntos de columnas correspondientes a los sustantivos involucrados en el tratamiento.

**Tabla 4.18.** Conjunto de columnas y tablas de los sustantivos *nombre*, *ciudad* y *compañía*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<b>nombre</b>	Categories.CategoryName Customers.CompanyName Customers.ContactName	$\emptyset$

	Employees.LastName Empleyes.FirstName Orders.ShipName Products.ProductName Shippers.CompanyName Suppliers.CompanyName Suppliers.ContactName	
<b>ciudad</b>	Customers.City Employees.City Orders.City Suppliers.City	∅
<b>compañía</b>	Customers.CompanyName Shippers.CompanyName Suppliers.CompanyName	∅

La tabla a la que pertenece la columna obtenida para el valor de la condición es *Suppliers* como se observa en la tabla 4.18, de los sustantivos involucrados en esta consulta (*nombre*, *ciudad* y *compañía*) se seleccionan las columnas cuya tabla sea la misma que genere la frase *Where*.

La columna *Suppliers.CompanyName* es la que se obtuvo en la condición, por lo tanto, las columnas *Supplier.ContactName* y *Suppliers.City* son las columnas probables que se solicitan en la condición. Aunque no se logra llegar a una sola columna este resultado si incluye la solicitud que corresponde a la frase *Select* de la consulta, por lo que de esta manera se está dando respuesta a la petición del usuario.

# Capítulo 5

## IMPLEMENTACIÓN

---

En este capítulo se muestra el análisis realizado al código del traductor de la ILNBD del ITCM, destacando los algoritmos principales del traductor que son los que conforman las fases de configuración y de traducción de la interfaz. También se muestra el algoritmo que fue modificado para llevar a cabo la implementación.

### 5.1 Análisis de la implementación del traductor del ITCM

El traductor esta implementado en el lenguaje Java, utiliza algoritmos para la creación de los metadatos, el diccionario de dominio y un algoritmo general de traducción. En la tabla 5.1 se listan los algoritmos implementados en la interfaz y su descripción.

**Tabla 5.1.** Algoritmos principales de la interfaz

ALGORITMO	OBJETIVO
CrearMetadatos	Crea el diccionario de metadatos
CrearDominio	Crea el diccionario de dominio
Algoritmo General	Algoritmo principal del traductor

El algoritmo general involucra una serie de algoritmos los cuales se listan en la tabla 5.2.

**Tabla 5.2.** Algoritmos derivados del Algoritmo General

ALGORITMO	OBJETIVO
Preprocesamiento	Algoritmo encargado de realizar un preprocesamiento a la consulta en lenguaje natural del usuario
SeparacionDeConsulta	Algoritmo encargado de identificar que partes de la oración corresponden a las frases <i>SELECT-FROM</i> Y <i>WHERE</i>
TratamientoPreposicionesConjunciones	Algoritmo que realiza el tratamiento especial aplicado a las preposiciones y conjunciones
Eliminar2preposicion2	Eliminar el caso valor numérico 2-preposición-valor numérico 2
ConvertirPostFijo	Fragmento de la oración que corresponde a la frase <i>SELECT-FROM</i>
RealizarOperacionesUnionInterseccion	En este algoritmo se llevan a cabo las operaciones de unión e intersección
Intersección	Se realiza la intersección de conjuntos
Unión	Se realiza la unión de conjuntos
FaseDeMejoramiento	Se lleva a cabo la fase de

	mejoramiento
TratamientoParteWHERE	Aplicar un tratamiento especial a la parte de la oración que corresponde a la cláusula <i>WHERE</i> .
Grafo	Se crea un grafo a partir de las columnas y tablas marcadas
FormarConsultaSQL	Formar la consulta SQL a partir de la información obtenida del subgrafo
PresentarResultado	Presentar el resultado al usuario

Durante la revisión de los algoritmos involucrados en la interfaz se realizó también la revisión del código de cada algoritmo en los cuales se utiliza la librería *java.sql* [48]. Esta es una interfaz de programación de aplicaciones API que contiene el conjunto de funciones y métodos para acceder y procesar información almacenada en bases de datos [49] usando el lenguaje de programación Java.

## 5.2 Diseño del tratamiento de la preposición “de” para más de dos sustantivos

En este punto se retomará la problemática con el tratamiento de la preposición “de” que es la situación anterior en la que se encontraba la interfaz y se explicará el proceso teórico de solución al problema para obtener una mejora en el porcentaje de éxito.

a) Problemática en el Tratamiento de la Preposición “de”

### Ejemplo 1:

*Dame el nombre del contacto de la compañía de nombre “Alfreds Futterkiste”*

Para esta consulta el tratamiento de preposiciones realiza las siguientes operaciones:

$$\textit{nombre} \cap \textit{contacto} \cap \textit{compañía} \cap \textit{nombre}$$

Al obtener como resultado final un conjunto vacío provocaba que no hubiera una traducción de la consulta, lo cual no ayudaba a mejorar el desempeño de la interfaz.

Esta situación se presentaba con este tipo de consultas en las que por medio de la preposición “de” se unían diferentes sustantivos incluyendo los que indican a qué entidad pertenece la información que se está solicitando: “*nombre del contacto de la compañía de nombre*”.

Para resolver esta problemática, se propuso una estrategia que mejoraría el tratamiento de preposiciones existente para contemplar este tipo de consultas.

b) Propuesta de solución diseñada

**Ejemplo1:**

*Dame en nombre del contacto de la compañía de nombre “Alfredo Futterkiste”*

**Tabla 5.3.** Columnas resultantes de la intersección de los sustantivos Nombre y Contacto

Intersección	Resultado
nombre $\cap$ contacto	Customers.ContactName Suppliers.ContactName

En la tabla 5.3 se muestra el conjunto resultante de la intersección entre los primeros 2 sustantivos, es diferente de vacío, sin embargo al seguir realizando las operaciones de intersección con el resto de los sustantivos obtenemos finalmente un conjunto vacío como se muestra en la tabla 5.4.

**Tabla 5.4.** Conjunto de columnas resultante de la intersección de los sustantivos *nombre, contacto, compañía y nombre*

Intersección	Resultado
nombre $\cap$ contacto $\cap$ compañía $\cap$ nombre	$\emptyset$

Anteriormente se proponía que se realizara la fase siguiente del proceso para determinar a qué columnas hacía referencia la condición pero atendiendo las recomendaciones del comité tutorial donde mencionaban que el análisis de la preposición “de” con respecto a la frase *Select* se separara del tratamiento de la frase *Where*, se optó por rescatar solamente los conjuntos resultantes entre los sustantivos que habían dado un resultado diferente de vacío para dejar ese conjunto como resultado en la fase del análisis de la frase *Select*.

En esta nueva consideración para los conjuntos resultantes vacíos existe el caso en el que ninguna intersección entre sustantivos da un resultado diferente de vacío, esto se muestra con el siguiente ejemplo.

**Ejemplo 2:**

Dame el nombre de la ciudad de la compañía “*Exotic Liquids*”

Se realizan las operaciones de intersección o unión, se guardan los conjuntos resultantes donde se obtuvieron columnas.

**Tabla 5.5.** Columnas resultantes de la intersección de los sustantivos *nombre y ciudad*

Intersección	Resultado
nombre $\cap$ ciudad	$\emptyset$

En la tabla 5.5 podemos observar que al hacer la intersección entre los primeros sustantivos obtenemos un conjunto vacío como resultado lo cual conlleva a

obtener como resultado final un conjunto vacío al intersectarse este conjunto resultante con el sustantivo *compañía*.

Se utilizan los conjuntos donde se obtuvieron columnas, en este caso no existen conjuntos resultantes de alguna intersección, entonces se utilizarán los conjuntos de columnas correspondientes a los sustantivos involucrados en el tratamiento.

**Tabla 5.6.** Conjunto de Columnas y Tablas de los sustantivos *nombre*, *ciudad* y *compañía*

Sustantivo	Conjunto de Columnas	Conjunto de Tablas
<b>nombre</b>	Categories.CategoryName Customers.CompanyName Customers.ContactName Employees.LastName Employees.FirstName Orders.ShipName Products.ProductName Shippers.CompanyName Suppliers.CompanyName Suppliers.ContactName	∅
<b>ciudad</b>	Customers.City Employees.City Orders.City Suppliers.City	∅
<b>compañía</b>	Customers.CompanyName Shippers.CompanyName Suppliers.CompanyName	∅

En el ejemplo anterior el conjunto de columnas resultante del análisis de la frase *Select* serían todas las columnas involucradas con cada uno de los sustantivos de la frase.

### 5.3 Implementación del diseño propuesto para el tratamiento de la preposición “de”

En este punto se desglosa el algoritmo que fue modificado con la implementación. Se muestra el algoritmo cómo funcionaba anteriormente [2] y cómo funciona después de la modificación que incluye la estrategia propuesta.

a) Algoritmo antes de los cambios hechos en la implementación

**Nombre del algoritmo:**

RealizarOperacionesUnionInterseccion.

**Parámetros de entrada:**

oracionPostFija: Fragmento de la oración que corresponde a la frase *SELECT – FROM* en orden postfijo de acuerdo a los valores numéricos de las preposiciones y conjunciones.

**Devuelve:**

Nada

**Objetivo:**

En este algoritmo se llevan a cabo las operaciones de unión e intersección.

**Algoritmo:**

```

01. Inicio
02.   Pila = vacío
03.   num = 1
04.   Para cada palabra en oracionPostFija
05.     Si palabraActual es preposición o conjunción
06.       Si la Pila no está vacía
07.         conjuntoResultado = "conjunto" + num
08.         operador2 = Pila.Pop
09.         operador1 = Pila.Pop
10.       Si palabraActual es preposición
11.         Interseccion(operador1, operador2, conjuntoResultado)
12.       De lo Contrario
13.         Union(operador1, operador2, conjuntoResultado)
14.       Fin Si
15.       Pila.Push(conjuntoResultado)
16.       num++
17.     Fin Si

```

```

18.         De lo Contrario
19.         Pila.Push(palabraActual)
20.         Fin Si
21.     Fin Para
22.     nombreTabla = Pila.Pop
23.     Insertar en la tabla resultadoColumnas la información de la tabla
        nombreTablaColumnas
24.     Insertar en la tabla resultadoTablas la información de la tabla
        nombreTablaTablas
25.     BorrarTablasAuxiliares
26. Fin Inicio

```

**Observaciones:**

Aquí se resuelve la oración postfija para obtener un conjunto de columnas y de tablas. Las conjunciones y preposiciones serán los operandos y las palabras serán los operadores. Las preposiciones realizan una intersección (línea 11) y las conjunciones una unión (línea 13). El último elemento al final de la pila es el nombre de la tabla temporal que contiene el conjunto de columnas y tablas (línea 22), el conjunto de columnas se guarda en la tabla *resultadoColumnas* (línea 23) y el conjunto de tablas se guarda en *resultadoTablas* (línea 24), después se eliminan todas las tablas temporales (línea 25).

b) Algoritmo con los cambios hechos en la implementación

**Nombre del algoritmo:**

RealizarOperacionesUnionInterseccion.

**Parámetros de entrada:**

oracionPostFija: Fragmento de la oración que corresponde a la frase *SELECT – FROM* en orden postfijo de acuerdo a los valores numéricos de las preposiciones y conjunciones.

**Devuelve:**

Nada

**Objetivo:**

En este algoritmo se llevan a cabo las operaciones de unión e intersección.

**Algoritmo:**

```

01. Inicio
02.   Pila = vacío
03.   PilaAux = vacío
04.   num = 1
05.   Para cada palabra en oracionPostFija
06.     Si palabraActual es preposición o conjunción
07.       Si la Pila no está vacía
08.         conjuntoResultante = "conjunto" + num
09.         operador2 = Pila.Pop
10.         operador1 = Pila.Pop
11.         Si palabraActual es preposición
12.           Interseccion(operador1, operador2, conjuntoResultante)
13.         De lo Contrario
14.           Union(operador1, operador2, conjuntoResultante)
15.         Fin Si
16.         Pila.Push(conjuntoResultante)
17.         PilaAux.Push(conjuntoResultante)
18.         num++
19.       Fin Si
20.     De lo Contrario
21.       Pila.Push(palabraActual)
22.     Fin Si
23.   Fin Para
24.   Para cada nombre de tabla en la pila de resultados PilaAux
25.     nombreTablaActual = PilaAux.Pop
26.     Insertar en la tabla resultadoColumnas la información de la tabla
     nombreTablaActual Columnas
27.     Insertar en la tabla resultadoTablas la información de la tabla
     nombreTablaActual Tablas
28.     si las tablas de resultados tienen elementos
29.       break
30.   Fin Para
31.   Si tablas de resultados no tienen elementos
32.     Para cada palabra en oracionPostFija
33.       Si palabraActual es diferente de preposición && conjunción
34.         Insertar en la tabla resultadoColumnas la información de la
         tabla palabraActual Columnas
35.         Insertar en la tabla resultadoTablas la información de la
         tabla palabraActual Tablas
36.       Fin Si
37.     Fin Para
38.   Fin Si
39. Fin Inicio

```

**Observaciones:**

Aquí se resuelve la oración postfija para obtener un conjunto de columnas y de tablas. Las conjunciones y preposiciones serán los operandos y las palabras

serán los operadores. Las preposiciones realizan una intersección (línea 12) y las conjunciones una unión (línea 14). Al final se revisan los elementos de la pila auxiliar los cuales son el nombre de las tablas temporales que contienen el conjunto de columnas y tablas (línea 24). El conjunto de columnas se guarda en la tabla *resultadoColumnas* (línea 26) y el conjunto de tablas se guarda en *resultadoTablas* (línea 27). Si las tablas de resultados no tienen elementos se obtienen los conjuntos de columnas y tablas correspondientes a cada sustantivo de la frase *SELECT* (línea 33), el conjunto de columnas se guarda en la tabla *resultadoColumnas* (línea 34) y el conjunto de tablas se guarda en *resultadoTablas* (línea 35).

# Capítulo 6

## RESULTADOS Y CONCLUSIONES

---

Capítulo 6. En este capítulo se muestran las pruebas realizadas con la interfaz una vez teniendo finalizada la etapa de implementación. Se ejemplifican dos tipos de consultas que resuelve esta nueva implementación para el tratamiento de la preposición “*de*”, también se muestran los resultados para los corpus de las bases de datos Northwind, Pubs y ATIS. Finalmente, se presentan las conclusiones a las que se llegó en este trabajo así como también se proponen trabajos futuros y diversas propuestas que ayuden a mejorar la ILNBD para lograr una mayor traducción de consultas.

## 6.1 Pruebas

En esta sección se muestran las traducciones y resultados obtenidos con la interfaz, utilizando las consultas mostradas como ejemplo en el capítulo 4 de esta tesis.

### Ejemplo 1

#### Tipo de consulta:

Consulta con más de dos sustantivos en la frase *Select*.

#### Objetivo:

El usuario desea saber el nombre del contacto de una compañía específica.

#### Consulta en lenguaje natural:

*Dame en nombre del contacto de la compañía de nombre "Alfreds Futterkiste"*

#### Consulta SQL obtenida:

```
SELECT Customers.ContactName
FROM Customers
WHERE ((Customers.CompanyName LIKE 'Alfreds_Futterkiste'))
```

#### Resultado:

Customers.ContactName
Maria Anders

Este primer ejemplo representa al tipo de consultas donde existen más de dos sustantivos que deben intersectarse en la frase *Select* y que alguno de sus conjuntos resultantes intermedios es diferente de vacío. En el tratamiento de preposiciones que se manejaba anteriormente se obtenía un conjunto final vacío, sin embargo, existen conjuntos intermedios resultantes que son diferentes de

vacío, este ejemplo muestra la nueva implementación para el tratamiento de la preposición, “de” donde se toma como resultado final aquel conjunto existente el cual es diferente de vacío.

**Ejemplo 2**

**Tipo de consulta:**

Consulta con más de dos sustantivos en la frase *Select*.

**Objetivo:**

El usuario desea saber el nombre de la ciudad de una compañía específica.

**Consulta en lenguaje natural:**

*Dame el nombre de la ciudad de la compañía “Exotic Liquids”*

**Consulta SQL obtenida:**

```
SELECT Suppliers.City, Suppliers.CompanyName, Suppliers.ContactName
FROM Suppliers
WHERE ((Suppliers.CompanyName LIKE 'Exotic_Liquids'))
```

**Resultado:**

Suppliers.City	Suppliers.CompanyName	Suppliers.ContactName
London	Exotic Liquids	Charlotte Cooper

Este segundo ejemplo representa al tipo de consultas donde existen más de dos sustantivos que deben intersectarse en la frase *Select* y que ninguno de sus conjuntos resultantes intermedios es diferente de vacío. En el tratamiento de preposiciones que se manejaba anteriormente se obtenía un conjunto final vacío y con este se continuaba el proceso de traducción, ahora la nueva implementación para el tratamiento de la preposición “de” no toma ese conjunto final resultante ya

que es vacío y como no existe un conjunto intermedio resultante que sea diferente de vacío se toman como conjunto final todas aquellas tablas que estén involucrados con cada uno de los sustantivos de la frase *Select*.

## 6.2 Resultados obtenidos para el Corpus de la Base de Datos Northwind

**Tabla 6.1.** Resultados obtenidos con el corpus para la base de datos Northwind (González 2005)

<b>NORTHWIND</b>	<b>CORRECTAS</b>		<b>INCORRECTAS</b>	
Consultas	45	90%	5	10%
Sin traducción			2	
Traducidas	45		3	

**Tabla 6.2.** Resultados obtenidos con el corpus para la base de datos Northwind (González 2010)

<b>NORTHWIND</b>	<b>CORRECTAS</b>		<b>INCORRECTAS</b>	
Consultas	46	92%	4	8%
Sin traducción			1	
Traducidas	46		3	

Para el corpus correspondiente a la base de datos Northwind, se logro un incremento en el porcentaje de éxito de un 2%, debido a que el corpus no tiene un número de consultas considerable que pertenezcan al tipo de problema resuelto en este trabajo, el incremento no se muestra tan radical.

### 6.3 Resultados Obtenidos para el Corpus de la Base de Datos Pubs

**Tabla 6.3.** Resultados obtenidos con el corpus para la base de datos Pubs  
(González 2005)

<b>PUBS</b>	<b>CORRECTAS</b>		<b>INCORRECTAS</b>	
Consultas	36	72%	14	28%
Sin traducción			4	
Traducidas	36		10	

**Tabla 6.4.** Resultados obtenidos con el corpus para la base de datos Pubs  
(González 2010)

<b>PUBS</b>	<b>CORRECTAS</b>		<b>INCORRECTAS</b>	
Consultas	36	72%	14	28%
Sin traducción			4	
Traducidas	36		10	

Para el corpus correspondiente a la base de datos Pubs, no se logró un incremento en el porcentaje de éxito, ya que este corpus no tenía consultas que pertenecieran al tipo de problema resuelto en este trabajo.

### 6.4 Resultados Obtenidos para el Corpus de la Base de Datos ATIS

**Tabla 6.5.** Resultados obtenidos con el corpus para la base de datos ATIS  
(González 2005)

<b>ATIS</b>	<b>CORRECTAS</b>		<b>INCORRECTAS</b>	
Consultas	43	61.4%	27	38.6%
Sin traducción			5	
Traducidas			22	

**Tabla 6.6.** Resultados obtenidos con el corpus para la base de datos ATIS  
(González 2010)

<b>ATIS</b>	<b>CORRECTAS</b>		<b>INCORRECTAS</b>	
Consultas	46	65.7%	24	34.3%
Sin traducción			3	
Traducidas	46		21	

Para el corpus correspondiente a la base de datos ATIS, se logro un incremento en el porcentaje de éxito de un 4.3%, en este corpus tal como en el corpus para la base de datos Northwind, tampoco tiene un número de consultas considerable que pertenezcan al tipo de problema resuelto en este trabajo, sin embargo el incremento en el porcentaje de éxito de estos dos corpus prueban que si existieran más casos pertenecientes al que se resuelve, la interfaz sería capaz de contestar correctamente.

## 6.5 Conclusiones

Las interfaces de lenguaje natural para acceso a bases de datos cuentan con grandes ventajas para los usuarios que requieren de herramientas cada vez más útiles para el manejo de información. Por esta razón, es necesario que se continúe con el desarrollo de las interfaces.

En el ITCM se había desarrollado una interfaz con una técnica de traducción que trataba las preposiciones específicamente las “de”, esta técnica no era muy robusta ya que no abarcaba en su totalidad los casos presentados entre las preposiciones y los sustantivos, lo cual viéndolo como área de oportunidad para hacer crecer el rendimiento de la interfaz se consideró para este trabajo.

En este trabajo se diseñó una técnica para ampliar el tratamiento de la preposición “de” ya existente. Analizando detalladamente el funcionamiento interno de la interfaz pudimos enfocar cada uno de los algoritmos involucrados, los cuales hacen posible que se lleve a cabo el proceso de traducción de una consulta.

En el estudio del comportamiento de cada algoritmo involucrado en la interfaz, identificamos aquel donde se realizan las funciones necesarias para hacer posible el tratamiento de preposiciones, tal algoritmo fue modificado con el objetivo de lograr una mejora en el porcentaje de éxito de la interfaz.

Los cambios realizados al algoritmo, nos permiten obtener un resultado de aquellas consultas donde se realizaban intersecciones entre más de dos sustantivos obteniendo de estas una traducción a SQL y una respuesta en lenguaje natural favorable para el usuario, ya que con el algoritmo anterior no era posible tener al menos una traducción al lenguaje SQL con este tipo de consultas y por tal motivo tampoco se obtenía una respuesta en lenguaje natural para el usuario. Además es importante destacar que los resultados existentes obtenidos con el algoritmo original no fueron alterados, es decir que la interfaz sigue respondiendo correctamente aquellas consultas que ya traducía correctamente.

Las pruebas se llevaron a cabo con tres corpus de distinto dominio, los cuales son Northwind, Pubs y ATIS, de las cuales se obtuvo un incremento en Northwind de un 2% y en ATIS un 4.3% de éxito. Es importante mencionar que para obtener resultados más notables en el incremento del porcentaje de éxito, se debe considerar la existencia de consultas que presenten el tipo de problema que se ha solucionado en este trabajo.

## 6.6 Trabajos futuros

Diseñar e implementar un sistema de diálogo con el usuario para clarificar los requerimientos donde se demande información implícita.

Diseñar una técnica que permita que la ILNBD sea capaz de identificar cuando un dato no existe en la base de datos y pueda hacerle saber al usuario la ausencia de tal dato.

Diseñar e implementar técnicas para el tratamiento de adjetivos y verbos.

Adicionar mecanismos para la traducción correcta de consultas para mejorar las fases de traducción de la ILNBD que lo requieran.

## REFERENCIAS

- [1] Rodríguez, Myriam; “Diseño de una Técnica para la Traducción de Consultas con Información Implícita a una Base de Datos”. Tesis de Maestría. Instituto Tecnológico de Cd. Madero. 2005.
- [2] Silva, Omar; “Implementación de un Traductor de Lenguaje Natural Independiente de Dominio”. Tesis de Maestría. Instituto Tecnológico de Cd. Madero. 2005.
- [3] Cruz, Irma; “Evaluación de una Interfaz de Lenguaje Natural a Bases de Datos en Idioma Español”. Tesis de Maestría. Instituto Tecnológico de Cd. Madero. 2006.
- [4] Manning, Christopher D.; Schütze, Hinrich, “Foundations of Statistical Natural Language Processing”. Massachusetts Institute of Technology, 2001.
- [5] Daniel Jurafsky & James H. Martin, “Speech and Language Processing”. University of Colorado, Boulder, 2000.
- [6] Barton, G. Edward; Berwick, Robert Cregar; and Ristad, Eric Sven, “Computational Complexity and Natural Language”. The MIT Artificial Intelligence Laboratory, 1987.
- [7] Rojas J.C. Administrador de Diálogo para una Interfaz de Lenguaje Natural a Bases de Datos, Tesis de Doctorado, Depto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2009.
- [8] “Concepto de Lenguaje Natural”.  
[http://es.wikipedia.org/wiki/Lenguaje\\_natural](http://es.wikipedia.org/wiki/Lenguaje_natural).2009.
- [9] “Concepto de Lenguaje Formal”.  
[http://es.wikipedia.org/wiki/Lenguaje\\_formal](http://es.wikipedia.org/wiki/Lenguaje_formal).2009.
- [10] Liddy D. Natural Language Processing for Information Retrieval & Knowledge Discovery, School of Information Studies, Syracuse University, 2001.
- [11] Elmasri, R. Sistemas de bases de datos, Addison-Wesley, 1997.

- [12] Androutsopoulos I., Ritchie G.D., Thanisch P. Natural Language Interfaces to Databases - An Introduction. Natural Language Engineering, Vol 1:29-81, 1995.
- [13] Silberschatz S., Korth, y Sudarshan, Fundamentos de Bases de Datos, 5ta edición, McGraw- Hill, 2006.
- [14] C.J. Date, Introducción a los Sistemas de Bases de Datos, 7ª. ed., México, Prentice Hall, pp. 151-152, 2001.
- [15] Cimiano P., Haase P. y Heizmann J. Porting natural language interfaces between domains: an experimental user study with the ORAKEL system. In Proceedings of the 12th international conference on Intelligent user interfaces, Honolulu, Hawaii, Estados Unidos, pp. 180-190, 2007.
- [16] Boonjing V. y HSU C. "A New Feasible Approach to Natural Language Database Query", International Journal on Artificial Intelligence Tools, Vol. 20, No. 10, pp. 1-8, 2006.
- [17] Let. "A Frame-based Approach to Text Generation", The 21st Pacific Asia Conference on Language, Information and Computation: Proceedings, Vol. 21, pp. 192-201, 2007.
- [18] Djahantighi F. "Using Natural Language Processing in Order to Create SQL Queries", Proceedings of the International Conference on Computer and Communication Engineering 2008, Vol. 13, No. 15, pp. 600-604, 2008.
- [19] Ahmad A. "Efficient Transformation of a Natural Language Query to SQL for Urdu", Proceedings of the Conference on Language & Technology 2009, pp. 53-60, 2009.
- [20] Kovács L. "SQL Generation for Natural Language Interface". Journal of Computer Science and Control Systems. Vol. 2, No. 18, pp. 19-22, 2009.
- [21] Sarhan A. "A proposed architecture for dynamically built NLIDB systems". International Journal of Knowledge Based Intelligent Engineering Systems. Vol. 13, No. 2, pp. 59-70, 2009.
- [22] Rao G. "Natural Language Query Processing Using Semantic Grammar".

- International Journal on Computer Science and Engineering. Vol. 2, No. 2, pp. 219-223, 2010.
- [23] Zettlemoyer L. *Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorical Grammars*. In Proceedings of the Twenty First Conference on Uncertainty in Artificial Intelligence (UAI-05), pp. 658-666, 2005.
- [24] Bernstein A., Kaufmann E. y Kaiser C., Querying the Semantic Web with Ginseng: A Guided Input Natural Language Search Engine, in: 15th Workshop on Information Technologies and Systems (WITS 2005), Las Vegas, NV, 2005.
- [25] Lopez V., Pasin M. y Motta E. AquaLog: An Ontology-Portable Question Answering System for the Semantic Web. A. Gómez-Pérez and J. Euzenat (Eds.): ESWC 2005, LNCS 3532, Springer-Verlag Berlin Heidelberg 2005, pp. 546 -562, 2005.
- [26] Lopez V., Motta E. y Uren V. PowerAqua: Fishing the Semantic Web. Proceedings of the European Semantic Web Conference, ESWC 2006.
- [27] Kaufmann E, Bernstein A. y Zumstein R. Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. 5th International Semantic Web Conference (ISWC 2006), Athens, GA, 2006.
- [28] Kaufmann E., Bernstein A. y Fischer L. NLP-Reduce: A "naïve" but Domain-independent Natural Language Interface for Querying Ontologies, in: 4th European Semantic Web Conference (ESWC 2007), Innsbruck, A, 2007.
- [29] Li Y. DaNaLIX: a Domain-adaptive Natural Language Interface for Querying XML. Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pp. 1165-1168, 2007.
- [30] Wang C., Xiong M. y Zhou Q. Panto: A Portable Natural Language Interface to Ontologies, Lecture Notes in Computer Science. Vol. 4519/2007, pp. 473-487, 2007.
- [31] Lapoon T. Using a Machine Learning Approach for Building Natural

- Language Interfaces for Databases: Application of Advanced Techniques in Inductive Logic Programming. Journal of Computer Science & Informatics & Electrical Engineering. Vol. 2, No. 1, 2008.
- [32] Damljanovic D., Agatonovic M. y Cunningham H. Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. In Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010), Springer Verlag, Heraklion, 2010.
- [33] G. Hendrix, E. Sacerdoti, D. Sagalowicz, y J. Slocum. "Developing a Natural Language Interface to Complex Data." ACM Transactions on Database System, 1978.
- [34] D.L. Waltz. "An English Language Question Answering System for a Large Relational Database". Communications of the ACM, 1978.
- [35] F. Damerau, "Operating statistics for the transformational question answering system. American Journal of Computational question answering system". American Journal of Computational Linguistics, Volume 7, pág. 30 – 42, 1981.
- [36] Popescu A.M., et. al. Modern natural language interfaces to databases: composing statistical parsing with semantic tractability. In Proceedings of the 20th international conference on Computational Linguistics, 2004.
- [37] R. A. Bhootra, Natural Language Interfaces: Comparing English Language Front End and English Query, tesis de maestría, Virginia Commonwealth University, Diciembre, 2004.
- [38] González, Juan Javier; "Traductor de Lenguaje Natural Español a SQL para un Sistema de Consultas a Bases de Datos". Tesis Doctoral. Centro Nacional de Investigación y Desarrollo Tecnológico. 2005.
- [39] Domínguez, Ana Patricia; "Implementación de un Analizador Gramatical del Lenguaje Español". Tesis de Maestría. Instituto tecnológico de Ciudad Madero. 2002.

- [40] Alarcón, Erika; “Diseño de la Interfaz de Lenguaje Natural para Consultas a Bases de Datos”. Tesis de licenciatura. Instituto Tecnológico de Ciudad Madero. 2002.
- [41] Mendoza, Alejandro; “Desarrollo de un Módulo de Predicados Lógicos para Consultas a la Base de Datos de Alumnos del ITCM”. Tesis de licenciatura. Instituto Tecnológico de Ciudad Madero. 2002.
- [42] M.Smith, K. Castellanos; “Consultas a una Base de Datos en Lenguaje Natural Utilizando Múltiples Tablas”. Tesis de licenciatura. Instituto Tecnológico de Ciudad Madero. 2003.
- [43] Alarcón, Erika; “Diseño de una Técnica Basada en Grafos Semánticos para la Traducción de Consultas de Lenguaje Natural a Lenguaje Formal”. Tesis de Maestría. Instituto Tecnológico de Ciudad Madero. 2004.
- [44] Mendoza, Alejandro; “Construcción de un Preprocesador de Consultas en Lenguaje Natural a una Base de Datos”. Tesis de Maestría. Instituto Tecnológico de Ciudad Madero. 2004.
- [45] Delgado, José Francisco; “Representación de una Consulta de Lenguaje Natural a través de un Grafo”. Tesis de Licenciatura. Instituto Tecnológico de Ciudad Madero. 2005.
- [46] Torres, Yuridia; “Construcción de un Diccionario de Valores para una Interfaz de Lenguaje Natural para Bases de Datos”. Proyecto de titulación para Licenciatura. Instituto Tecnológico de Ciudad Madero. 2005.
- [47] Juárez, María Dolores; “Memoria para una Interfaz de Lenguaje Natural a Base de Datos”. CIIC’10 International Congress on Computer. Instituto Tecnológico de Ciudad Madero, 2010.
- [48] <http://download-llnw.oracle.com/javase/1.5.0/docs/api/java/sql/package-summary.html>
- [49] C.J. Date, Introducción a los Sistemas de Bases de Datos, 7ª. ed., México, Prentice Hall, 2001.