

División De Estudios De Posgrado E Investigación



TESIS

**IMPLEMENTACIÓN DE UN ADMINISTRADOR DE DIÁLOGO  
PARA UNA INTERFAZ DE LENGUAJE NATURAL PARA  
CONSULTAS A BASES DE DATOS**

Que para obtener el grado de:

**Maestro en Ciencias Computacionales**

Presenta:

**Rafael Ortega Cortez      G12072015**

Director de tesis:

*Dr. Rodolfo A. Pazos Rangel*

Ciudad Madero, Tamaulipas

Octubre 2014



"2014, Año de Octavio Paz"

Cd. Madero, Tamps; a **04 de Septiembre de 2014.**

OFICIO No.: U5.204/14  
AREA: DIVISIÓN DE ESTUDIOS  
DE POSGRADO E INVESTIGACIÓN  
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

**ING. RAFAEL ORTEGA CORTEZ**  
**NO. DE CONTROL G12072015**  
**PRESENTE**

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias de la Computación, el cual está integrado por los siguientes catedráticos:

PRESIDENTE :	DR. JOSÉ ANTONIO MARTÍNEZ FLORES
SECRETARIO :	DRA. LAURA CRUZ REYES
VOCAL :	DR. RODOLFO ABRAHAM PAZOS RANGEL
SUPLENTE :	M.C. JOSÉ APOLINAR RAMÍREZ SALDIVAR
DIRECTOR DE TESIS :	DR. RODOLFO ABRAHAM PAZOS RANGEL

Se acordó autorizar la impresión de su tesis titulada:

**"IMPLEMENTACIÓN DE UN ADMINISTRADOR DE DIÁLOGO PARA UNA INTERFAZ DE LENGUAJE NATURAL PARA CONSULTAS A BASES DE DATOS"**

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 Cineo*  
**M. P. MARÍA YOLANDA CHÁVEZ CINEO**  
**JEFA DE LA DIVISIÓN**



**S. E. P.**  
DIVISIÓN DE ESTUDIOS  
DE POSGRADO E  
INVESTIGACIÓN  
I T C M

c.c.p.- Archivo  
Minuta

MYCHC 'NICO' jar  
*M*



Ave. 1° de Mayo y Sor Juana I. de la Cruz, Col. Los Mangos, CP. 89440 Cd. Madero, Tam.  
Tel. (833) 357 48 20, Fax, Ext. 1002, e-mail: itcm@itcm.edu.mx

[www.itcm.edu.mx](http://www.itcm.edu.mx)



## DECLARACIÓN DE ORIGINALIDAD

Declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares.

Además, declaro que las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y publicaciones.

Además, en caso de infracción a los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director de tesis, así como al Instituto Tecnológico de Ciudad Madero y sus autoridades.



Rafael Ortega Cortez

## **AGRADECIMIENTOS**

A los miembros de mi comité tutorial de tesis, por sus observaciones y sugerencias en el desarrollo de este trabajo.

Al Dr. Rodolfo Abraham Pazos Rangel por el conocimiento compartido, el apoyo brindado y la paciencia. Por permitirme realizar este trabajo y hacerlo posible.

A los doctores que no formaron parte de mi comité pero que contribuyeron a mi formación en la maestría: Dr. Arturo Hernández Ramírez, Dr. Héctor Joaquín Fraire Huacuja, Dr. Juan Javier González Barbosa, Dra. Guadalupe Castilla Valdez.

A mi amigo Gerardo por toda su ayuda, y por animarme a hacer el intento de ingresar a la maestría.

A todas las personas con las que conviví en el ITCM, especialmente: Fanny, Yazmin, Eduardo, Oscar, Miguel, Javier.

## RESUMEN

Las interfaces de lenguaje natural para bases de datos permiten al usuario comunicarse con un sistema de información de una forma amigable; sin embargo, todavía se presentan dificultades en la traducción de la consulta en lenguaje natural a su equivalente en un lenguaje de consulta estructurada. Un ejemplo de estas problemáticas es la elipsis o economía de palabras.

El problema que se aborda en este trabajo es aclarar una consulta en lenguaje natural que presente una situación de elipsis por insuficiencia de conocimiento, por medio de un diálogo con el usuario. Para lograr esto se utilizó una tipificación de problemas en consultas y procesos de diálogo para cada caso de problemas.

Este proyecto de tesis se enfoca en la implementación de un administrador de diálogo que se acopló a una interfaz de lenguaje natural para consultas a bases de datos desarrollada en el Instituto Tecnológico de Ciudad Madero, con el propósito de mejorar el desempeño de la interfaz aumentando el porcentaje de acierto en las consultas.

# CONTENIDO

CAPÍTULO 1: INTRODUCCIÓN.....	1
1.1 Objetivos.....	2
1.2 Justificación y beneficios .....	3
1.3 Descripción del problema .....	5
1.4 Descripción de la complejidad del problema .....	6
1.5 Alcances y limitaciones.....	9
CAPÍTULO 2: MARCO TEÓRICO .....	11
2.1 Procesamiento de Lenguaje Natural.....	11
2.2 Interfaces de Lenguaje Natural.....	12
2.3 Consultas a bases de datos .....	14
2.4 Elipsis .....	16
2.5 Administrador de diálogo .....	18
2.6 Antecedentes.....	20
2.7 Trabajos relacionados .....	22
CAPÍTULO 3. METODOLOGÍA.....	26
3.1 Problemas en consultas .....	26
3.1.1 Problemas básicos en consultas.....	28
3.1.2 Problemas dobles en consultas .....	29
3.2 Estructura de datos.....	31
3.3 Algoritmos.....	34
3.3.1 Algoritmos de identificación de problemas.....	34
3.3.2 Algoritmos de los procesos de diálogo.....	39
3.4 Interfaces de usuario .....	44
CAPÍTULO 4. EXPERIMENTACIÓN .....	47
4.1 Corpus de consultas para prueba.....	47
4.2 Pruebas .....	48
4.2.1 Prueba de desempeño.....	49
4.2.2 Prueba comparativa .....	50

4.3	Resultados.....	51
CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS .....		53
5.1	Conclusiones .....	53
5.2	Trabajos futuros.....	54
ANEXO A. CORPUS DE CONSULTAS .....		56
ANEXO B. DESCRIPCIÓN DE LA BASE DE DATOS ATIS.....		66
ANEXO C. DESCRIPCIÓN DE LA BASE DE DATOS NORTHWIND .....		75
ANEXO D. DESCRIPCIÓN DE LA BASE DE DATOS GEOBASE .....		81
GLOSARIO.....		85
REFERENCIAS.....		86

## Lista de tablas

2.1 Descripción de algunos trabajos relacionados .....	22
2.2 Comparación de las ILNBDs de algunos trabajos relacionados .....	23
3.1 Tipificación de problemas considerando los elementos de una consulta a bases de datos.....	27
3.2 Tipificación de problemas básicos en consultas.....	28
3.3 Tipificación de problemas dobles .....	30
3.4 Representación de la estructura de datos de la ILNBD con una consulta procesada .....	32
3.5 Representación de la estructura de datos de la ILNBD con una consulta procesada de ejemplo con el problema 2.1 .....	38
3.6 Representación de la estructura de datos de la ILNBD con una consulta procesada de ejemplo con el problema 4.1 .....	41
3.7 Representación de la estructura de datos de la ILNBD con una consulta de ejemplo sin elipsis por insuficiencia de conocimiento .....	42
3.8 Elementos de una consulta de ejemplo.....	44
4.1 Composición del corpus de consultas de prueba según las bases de datos .....	47
4.2 Composición del corpus de consultas de prueba según los casos de prueba.....	48
4.3 Características del equipo .....	49
4.4 Características del software .....	49
4.5 Software utilizado en la prueba de comparación entre ILNBDs.....	50
4.6 Resultados resumidos de la prueba de comparación.....	51
4.7 Resultados resumidos de la prueba de comparación.....	51
4.8 Resultados de la prueba de comparación de la ILNBD (CENIDET).....	52
4.9 Resultados de la prueba de comparación del número de ciclos de diálogo ..	52
A.1 Consultas para pruebas con el problema 2.1.....	56
A.2 Consultas para pruebas con el problema 3.1.....	57
A.3 Consultas para pruebas con el problema 4.1.....	57

A.4 Consultas para pruebas con el problema 5.1.....	58
A.5 Consultas para pruebas con los problemas 2.1 y 3.1 .....	59
A.6 Consultas para pruebas con los problemas 2.1 y 4.1 .....	60
A.7 Consultas para pruebas con los problemas 2.1 y 5.1 .....	61
A.8 Consultas para pruebas con los problemas 4.1 y 3.1 .....	62
A.9 Consultas para pruebas con los problemas 4.1 y 5.1 .....	63
A.10 Consultas para pruebas con los problemas 3.1 y 5.1 .....	64

## Lista de figuras

1.1 Arquitectura actual de la ILNBD .....	5
1.2 Arquitectura de la ILNBD con administrador de diálogo.....	6
2.1 Una arquitectura general de una ILN.....	13
2.2 Flujo de una ILNBD .....	13
2.3 Arquitectura general de un sistema de diálogo.....	19
3.1 Tablas de una base de datos de ejemplo.....	43
3.2 Interfaz de usuario para el diálogo del problema 2.1.....	45
3.3 Interfaz de usuario para el diálogo del problema 3.1.....	45
3.4 Interfaz de usuario para el diálogo de los problemas 4.1 y 5.1.....	46
B.1 Esquema de la base de datos ATIS.....	66
C.1 Esquema de la base de datos Northwind .....	75
D.1 Esquema de la base de datos Geobase .....	81

## CAPÍTULO 1: INTRODUCCIÓN

En la actualidad, es más frecuente el uso de los sistemas computacionales para obtener información a partir de datos, con el propósito de tomar decisiones. Esto ocurre en diversos ámbitos: empresarial, industrial, académico y muchos más; además, en diferentes áreas de las organizaciones, no sólo en el departamento de tecnologías de información. Estas aplicaciones de acceso a la información son utilizadas incluso por usuarios sin conocimientos computacionales especializados, por lo que es necesario que sean de fácil operación.

El lenguaje natural hablado y escrito es un área de investigación en constante desarrollo. Este trabajo de tesis aborda el tema del lenguaje natural escrito aplicado a las interfaces de lenguaje natural para consulta a bases de datos (ILNBDs).

Las ILNBDs son desarrolladas con el propósito de facilitar al usuario el acceso a datos. Éstas se utilizan para obtener información almacenada en una base de datos a través de una solicitud formulada en lenguaje natural.

Además de las ILNBDs, también existen otros medios y métodos para acceder a la información contenida en bases de datos, tanto con ventajas como desventajas comparándolos entre sí. Por ejemplo, el lenguaje SQL (Structured Query Language) es un medio con alta capacidad para la formulación de consultas, pero se requiere de una capacitación especializada para su manejo adecuado. Por otro lado, las aplicaciones con ambientes gráficos y de fácil operación poseen limitaciones en la formulación de las consultas.

Debido a esta situación, las interfaces de lenguaje natural (ILNs) surgen como una solución, permitiendo su operación en manos de cualquier tipo de

usuario. Las ILNBDs son un tipo de ILN, cuyo objetivo es facilitar el acceso a la información almacenada de manera estructurada en bases de datos.

Las ILNBDs permiten a cualquier clase de usuario acceder a la información que busca, de una manera más sencilla; sin embargo, todavía se presentan algunos problemas que no se han resuelto satisfactoriamente con las metodologías aplicadas. Por lo tanto, el objetivo de esta tesis es continuar con el esfuerzo previamente realizado en una ILNBD [Aguirre, 2014] implementada en el ITCM, a través del diseño y desarrollo de procesos de diálogo escrito, para tratar de resolver el problema de las consultas con elipsis<sup>1</sup> por insuficiencia de conocimiento.

## 1.1 Objetivos

Objetivo general:

Mejorar el desempeño de una ILNBD [Aguirre, 2014] mediante la incorporación de procesos de diálogo que permitan proporcionar al usuario los resultados esperados.

Objetivos específicos:

1. Adaptar o elaborar una tipificación de problemas en consultas [Rojas, 2009], que incluya:
  - a) Consultas con elipsis por insuficiencia de conocimiento.
  - b) Consultas mal formuladas.
  - c) Consultas ilógicas para la semántica de la base de datos.

---

<sup>1</sup> Ver Glosario.

2. Diseñar y desarrollar los procesos de diálogo correspondientes al inciso (a) de los problemas presentados en el objetivo específico 1 considerando la independencia de dominio<sup>2</sup>.
3. Los procesos de diálogo deben permitir al usuario, mediante ciclos pregunta-respuesta, obtener la información solicitada de forma eficaz.
4. Los procesos de diálogo implementados deben ser incorporados a la ILNBD [Aguirre, 2014] desarrollada en el ITCM, para aumentar el porcentaje de acierto en las consultas.

## **1.2 Justificación y beneficios**

En la implantación de los sistemas de información en una organización, generalmente se considera un proceso de capacitación para la operación del mismo. Esta capacitación dependerá del tamaño y complejidad del sistema, y regularmente se imparte al operario y responsable de éste, dejando de lado a otros posibles usuarios: ejecutivos, gerentes, jefes de departamentos y otros. Debido a la necesidad del acceso a la información por parte de diversos miembros de la organización, es requisito indispensable en las herramientas de información que sean de fácil uso, proporcionen respuesta rápida y sean amigables con todos sus usuarios.

Las ILNBDs son una solución que responde a esta problemática, ofreciendo además, la capacidad de obtener información a través de consultas a bases de datos, sin que el usuario deba aprender un lenguaje de consulta como SQL.

---

<sup>2</sup> Ver Glosario.

Sin embargo, hasta el día de hoy, las ILNBDs siguen presentando problemas, como errores en la traducción de la consulta. Las consultas mal interpretadas ofrecerán resultados equivocados para el usuario, posiblemente causando una toma de decisiones incorrecta.

El proceso de traducción de la consulta es uno de los principales problemas en las ILNBDs. Traducir una consulta en lenguaje natural a su equivalente en un lenguaje de consulta de bases de datos, es un proceso que incluye problemas como: elipsis semántica, ambigüedad, contexto de la información, etc.

En este trabajo de tesis se muestra que, mediante la incorporación de un administrador de diálogo a una ILNBD [Aguirre, 2014] con independencia de dominio, el usuario puede obtener la respuesta esperada mediante ciclos pregunta-respuesta para aclarar una consulta que no pueda ser totalmente interpretada por una ILNBD.

Los beneficios que se obtienen con este proyecto de tesis son los siguientes:

1. En general, facilitar la consulta a bases de datos, incorporando procesos de diálogo que permiten al usuario, mediante ciclos de pregunta-respuesta, obtener los resultados esperados.
2. Específicamente, incrementar el porcentaje de respuestas correctas y confiables al usuario, reduciendo el margen de error al responder con información no requerida por el usuario.
3. Además, ya que la tipificación de problemas es general (aplicable a diversos lenguajes como inglés, francés e italiano, que son similares en estructura al español) y es independiente de dominio [Rojas, 2009], los procesos de diálogo tienen estas características también.

### 1.3 Descripción del problema

Las interfaces de lenguaje natural para bases de datos permiten al usuario comunicarse con un sistema de información de una forma más amigable y natural; sin embargo, todavía presentan algunas dificultades en la traducción de la consulta en lenguaje natural a su equivalente en un lenguaje de consulta de bases de datos, tales como: la elipsis semántica, las consultas formuladas incorrectamente, etc.

En este trabajo se aborda el problema de aclarar una consulta en lenguaje natural que presente una situación de elipsis por insuficiencia de conocimiento, por medio de la adición de procesos de diálogo con el usuario, para obtener los datos faltantes de la consulta mencionada y con ello completar la traducción a lenguaje de consulta de bases de datos, en la ILNBD independiente de dominio desarrollada por Marco Aguirre [Aguirre, 2014]. En la Figura 1.1 se muestra la arquitectura de la ILNBD, y en la Figura 1.2 se incluye en la arquitectura el módulo que administra los procesos de diálogo mencionados (Administrador de diálogo).

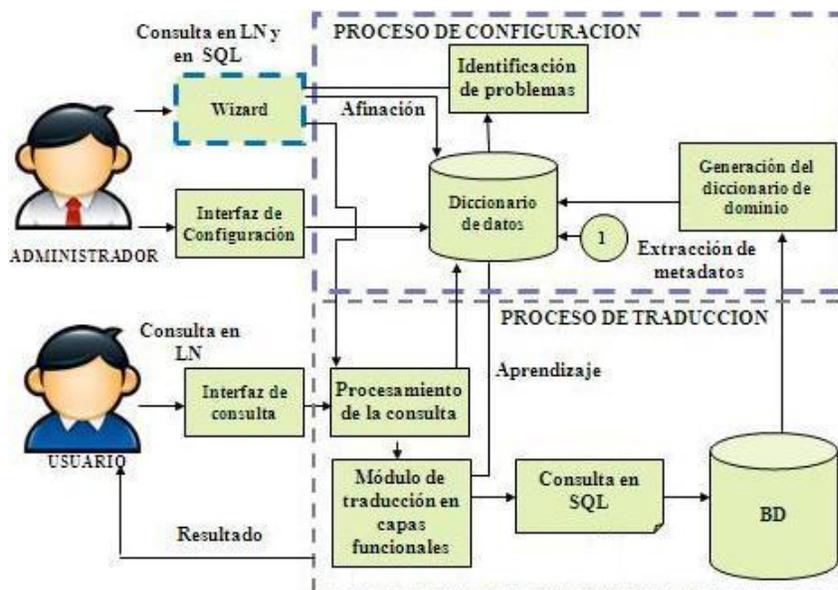


Figura 1.1 Arquitectura actual de la ILNBD

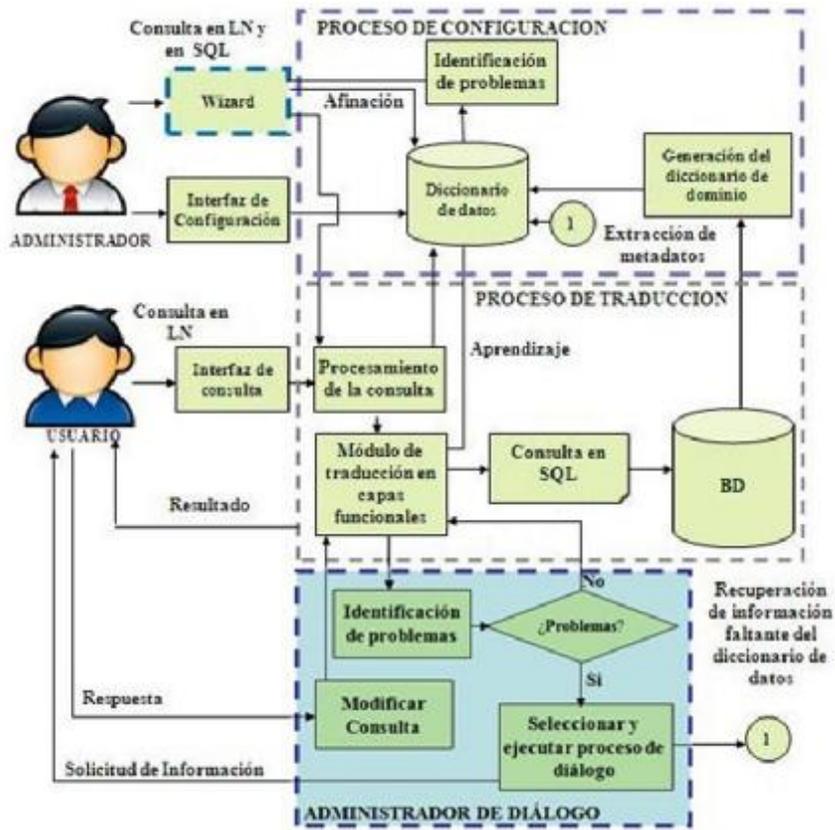


Figura 1.2 Arquitectura de la ILNBD con administrador de diálogo

## 1.4 Descripción de la complejidad del problema

Este trabajo de tesis se enfoca al problema de la elipsis por insuficiencia de conocimiento, también referida como economía de palabras. Aunque el problema se ha tratado de resolver por desarrolladores de sistemas e investigadores de trabajos anteriores en ILNBDs, no ha sido identificado de manera precisa y más aún, no ha sido resuelto de modo formal y sistemático, incluso se ha evadido pasando éste al administrador del sistema de la BD, o quien está a cargo de la configuración de la interfaz para adaptarla a una BD específica. Por ejemplo, en la consulta a una BD de animales: “how many mammals are there”, los documentos de ayuda del software ELF [ELF, 2002] sugieren al administrador de sistemas que

configure la interfaz de manera que sea capaz de convertir la consulta “how many mammals are there” a “how many animals are mammals”, para que el usuario obtenga resultados más adecuados.

La elipsis semántica se presenta generalmente en todas las conversaciones cotidianas, y pocas veces se requiere aclarar a qué se refiere una frase o expresión, ya que la mayoría de la información omitida suele estar implícita en el contexto. Sin embargo, en una ILNBD, la elipsis por insuficiencia de conocimiento puede ocasionar una interpretación errónea de una consulta, y por lo tanto, causar una traducción incorrecta de ésta. Considérese el siguiente ejemplo: “despliega las calificaciones de Juan Pérez”. Cualquier persona con conocimientos sobre BDs podría traducir esta solicitud de información a una consulta en SQL dado el esquema de la BD, pero para una ILNBD con independencia de dominio podría ser difícil de responder a esa solicitud. El problema es que la consulta no especifica la siguiente información: la tabla donde se almacenan los datos de las materias que cursó el estudiante, y la tabla y el campo donde se almacenan los nombres de los estudiantes. Si se edita la consulta original especificando la información faltante, entonces quedaría: “desplegar las calificaciones de las materias cursadas por el estudiante cuyo nombre es Juan Pérez”.

Cabe mencionar que los procesos de diálogo desarrollados en este trabajo de tesis, para afrontar la problemática de la elipsis por insuficiencia de conocimiento, deben funcionar para cualquier BD y con cualquier consulta, dentro del marco de las limitaciones especificadas. Esto significa que los procesos de diálogo deben poder aplicarse a un total de consultas que sería muy grande, pues se considerarían todas las BDs que existen y que se crearan en un futuro, así como todas las consultas posibles que se pudieran elaborar para cada una de esas BDs.

Aunado a lo anterior, se tiene el hecho de que el número de consultas que puede efectuarse a una base de datos en particular puede llegar a ser muy

grande; y por lo tanto, si se suman las consultas de todas las bases de datos relacionales habidas y por haber, el número total resultaría astronómico.

Partiendo del razonamiento anterior, no sería factible (por razones de tiempo) tomar cada una de las bases de datos relacionales existentes y tratar de implementar procesos de diálogo para cada una de éstas. Para ilustrar el problema considérese el caso de un sistema denominado BirdQuest. Esta interfaz fue desarrollada para un sitio web donde la gente mira programas de televisión de la naturaleza y puede hacer consultas relacionadas al programa, en este caso para aves nórdicas [Niesler, 2001]. Un ejemplo de diálogo de esta interfaz es el siguiente:

User: Information about Crows

System: Birdquest can give information about the following properties for crows: call, habitat, appearance, feed, breeding. Please specify what you're interested in.

User: I am interested in Call

System: Call is vague, BirdQuest can give information about the following aspects of call: general, warning, song. Please specify what you're interested in.

User: General, please.

System: BirdQuest found the following information about general call for the crow: "The crows croak, krah (sic), krah (sic), is well known"

En este punto es oportuno aclarar que BirdQuest, como muchas otras, es una ILNBD dependiente del dominio. Al observar el ejemplo de diálogo de BirdQuest, surge la siguiente pregunta: ¿se podrá adaptar el proceso de diálogo de BirdQuest para que pueda aplicarse a otras BDs (v.g., control escolar, historiales clínicos)? Al analizar el proceso de diálogo de BirdQuest, se observa que es muy dependiente del dominio; así que la respuesta a la pregunta es no.

En tales circunstancias, surge otra pregunta: ¿será posible diseñar un proceso de diálogo que pueda usarse con cualquier BD? Ya que el proceso de diálogo en cuestión debería poder aplicarse a cualquier BD, incluso aquéllas que

uno desconozca, de esta pregunta se derivan las siguientes: ¿será posible diseñar un proceso de diálogo para una BD desconocida? En la tesis de doctorado de Carlos Rojas se respondió afirmativamente a esta interrogante.

Sin embargo, el administrador de diálogo desarrollado por Rojas no es posible acoplarlo a la interfaz desarrollada por Aguirre, ya que esta última utiliza estructuras de datos complejas, para la consideración de consultas que incluyan sustantivos, verbos, adjetivos y cualquier preposición, a diferencia de la interfaz para la que Rojas desarrolló el administrador de diálogo, que posee la limitación de considerar sólo los sustantivos y la preposición “de” en las consultas del usuario.

## **1.5 Alcances y limitaciones**

Los alcances de este trabajo de tesis son los siguientes:

- a) Adaptación de una tipificación de tipos de problemas en consultas.
- b) Diseño de una tipificación de problemas dobles en consultas.
- c) Diseño de procesos de diálogo para los tipos de consulta con elipsis por insuficiencia de conocimiento.
- d) Implementación de ciclos pregunta-respuesta para los procesos de diálogo, permitiendo al usuario obtener la información solicitada.
- e) Los procesos de diálogo deben funcionar para la ILNBD desarrollada en el ITCM, los cuales deben poseer la característica de independencia de dominio.

Entre las limitaciones de este trabajo se encuentran las siguientes:

- a) El idioma soportado por el administrador de diálogo es únicamente español.
- b) El medio de comunicación del proceso de diálogo es en lenguaje escrito.
- c) No se trata el problema de la ambigüedad o preguntas mal formadas.

- d) Sólo se maneja un subconjunto de las consultas de SQL de la versión ISO/IEC 9075:1989(E) (SQL 1). Las funciones que no están implementadas son las relacionadas con operaciones de agrupación, inserción, borrado o actualización de información.
- e) No proporciona información que no esté explícita en la BDs, ya que no se manejan bases de datos deductivas.
- f) Debido a que en SQL una consulta se puede expresar de diferentes maneras, no se transforma una consulta a su equivalente optimizada.
- g) Las consultas deben ser léxica y sintácticamente correctas. El usuario debe considerar esta condición para obtener una respuesta satisfactoria.
- h) La información solicitada en la consulta a la base de datos debe expresarse antes que la condición.
- i) Las consultas pueden ser en forma interrogativa e imperativa.
- j) El formato para representar fechas debe ser: dd/mm/aaaa.
- k) Las BDs a consultar deben ser relacionales y deben permitir el acceso a los datos a través de un lenguaje de consulta estructurado.

## CAPÍTULO 2: MARCO TEÓRICO

Este capítulo presenta los conceptos principales relacionados con este trabajo de tesis, incluyendo también, los trabajos que fueron base fundamental para éste. Y por último, se resumen algunos trabajos relacionados.

### 2.1 Procesamiento de Lenguaje Natural

Algunos conceptos que se consideran importantes para facilitar la comprensión del tema de procesamiento de lenguaje natural son los siguientes.

#### *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 [Rojas, 2009]:

- § Los lenguajes naturales como español, inglés y francés.
- § Los lenguajes formales como los lenguajes de programación y el lenguaje de la lógica matemática.

#### *Lenguaje formal*

Un lenguaje formal es un lenguaje artificial creado por el hombre, el cual está formado por símbolos y fórmulas, y tiene como objetivo fundamental

formalizar la programación de computadoras o representar simbólicamente un conocimiento [Rojas, 2009].

### *Lenguaje natural*

Lenguaje hablado o escrito por humanos, opuesto a un lenguaje de programación utilizado para programar o comunicarse con computadoras. Existen dos campos en el estudio del entendimiento del lenguaje natural [Rojas, 2009]:

- § Entendimiento del lenguaje escrito, que utiliza el conocimiento léxico, sintáctico y semántico del lenguaje, unido a la información o conocimiento del dominio.
- § Entendimiento del lenguaje oral, que comprende todo lo del campo anterior junto con toda la fonología.

### *Procesamiento de lenguaje natural*

El procesamiento de lenguaje natural (PLN o NLP en inglés) es un conjunto de técnicas computacionales para analizar y representar naturalmente textos en uno o más niveles de análisis lingüísticos, con el fin de llevar a cabo el procesamiento del lenguaje como un humano para un rango de tareas y aplicaciones [Liddy, 1998].

## **2.2 Interfaces de Lenguaje Natural**

Las interfaces de lenguaje natural (ILNs) son mecanismos de comunicación entre una persona y una máquina a través de lenguaje natural. Por lo general, esta

comunicación es bidireccional, es decir, de tipo pregunta-respuesta. Una arquitectura general de una ILN se muestra en la Figura 2.1.

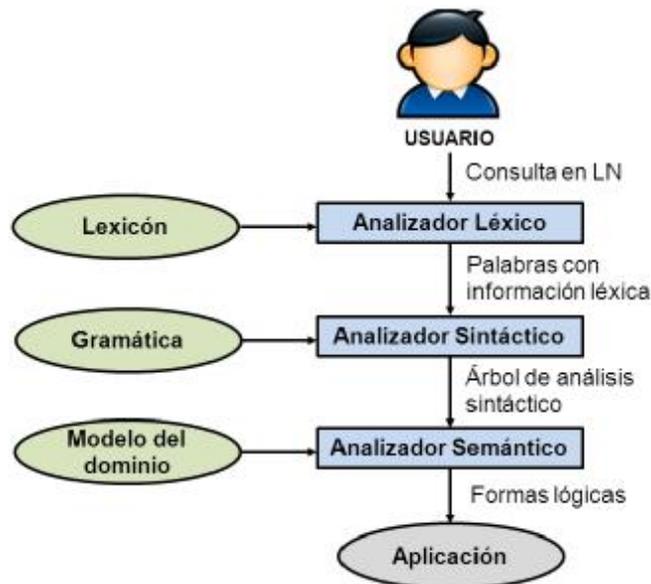


Figura 2.1. Una arquitectura general de una ILN

### *Interfaces de Lenguaje Natural para Bases de Datos*

Una Interfaz de Lenguaje Natural para Bases de Datos (ILNBD) es un sistema que permite al usuario acceder a la información almacenada en una base de datos formulando una solicitud en lenguaje natural [Androutsopoulos, 1995]. El flujo de una ILNBD se muestra en la Figura 2.2.

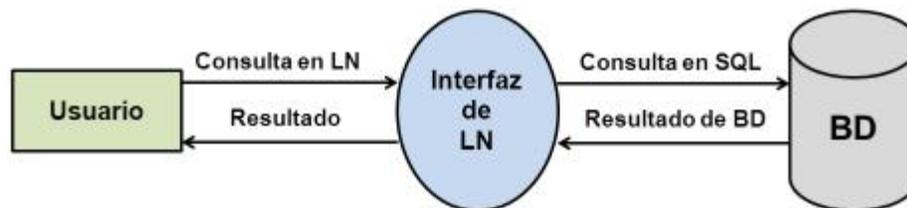


Figura 2.2. Flujo de una ILNBD

## 2.3 Consultas a bases de datos

Una consulta es el dictamen que por escrito o de palabra se pide o se da acerca de algo [RAE, 2014]. A partir de esta definición, se le llama consulta a base de datos a toda oración que está estructurada de manera interrogativa o imperativa y sirve para obtener información de alguna base de datos [Mellado, 2014]. Las consultas a bases de datos que se utilizan en este trabajo de tesis son introducidas en lenguaje natural a la ILNBD, para su procesamiento y traducción a un lenguaje especializado de consultas, en este caso, SQL.

### *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:2011. SQL es una combinación de constructores del álgebra relacional y del cálculo relacional<sup>3</sup>. Usando SQL es posible, además de definir la estructura de los datos, modificar los datos de la base de datos y especificar restricciones de seguridad [Silberschatz, 2006].

El lenguaje SQL tiene varios componentes [Rojas, 2009]:

- § Lenguaje de definición de datos (LDD). Proporciona comandos para la definición, borrado y modificación de los esquemas de tablas.

---

<sup>3</sup> Ver Glosario.

- § 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 tablas y a las vistas.

Una consulta a bases de datos se realiza con el comando SQL Select para extraer los datos de una o más tablas. La sintaxis básica de este comando es:

```
SELECT lista de columnas  
      FROM lista de tablas  
      [WHERE condiciones]
```

donde:

- § La cláusula SELECT especifica la(s) columna(s) a desplegar en el resultado de la consulta.
- § La cláusula FROM especifica la(s) tabla(s) que contiene(n) los datos que se solicitan en la consulta.
- § La cláusula WHERE especifica las condiciones que determinan cuáles renglones de la(s) tabla(s) deben incluirse en el resultado de la consulta. Esta cláusula es opcional, ya que sólo se incluye en el comando cuando se desea indicar un criterio de búsqueda en la consulta.

En una consulta en lenguaje natural que será traducida a una consulta en SQL, es necesario identificar dos secciones: la frase Select y la frase Where.

La frase Select está integrada por las unidades léxicas, de la consulta en lenguaje natural, que se requieren para ser traducidas a la cláusula Select en SQL. Al menos la frase Select debe estar presente en una consulta en lenguaje natural para que sea posible generar una consulta en SQL.

La frase Where está compuesta por las unidades léxicas que serán traducidas a la cláusula Where en SQL. La frase Where no es obligatoria en una consulta en lenguaje natural.

## 2.4 Elipsis

El término elipsis proviene del griego *elleipsis*, derivada de *elleipen*: omisión. Es la omisión de una o más palabras que son entendidas de manera obvia pero que deben ser especificadas para la construcción gramatical completa [Webster, 2003].

El fenómeno de la elipsis ha sido abordado desde una variedad de ángulos en los años recientes, y permanece difícil de clasificar [Smith, 2001]. En lingüística contemporánea se identifican diversos tipos de elipsis, entre ellos: pragmática, semántica, de frase verbal, de vacío, de drenado, basada en antecedente, basada en situación.

Dado que el tipo de elipsis a tratarse en este trabajo no puede ubicarse en alguno de los tipos mencionados, se propone una definición de *elipsis por insuficiencia de conocimiento*, basada en la de elipsis semántica, por lo que se presenta primero la definición de esta última.

### *Elipsis semántica*

La omisión de elementos que, aunque no son obligatorios sintácticamente, son requeridos para una completa interpretación semántica de un texto [McShane, 2004].

### *Elipsis por insuficiencia de conocimiento*

Tipo de elipsis semántica que ocurre en el ámbito de la interpretación de consultas en las ILNBDs. Es la omisión de una o más palabras en una consulta en lenguaje natural, que son necesarias para que una ILNBD elabore una traducción correcta a la expresión en SQL correspondiente.

A modo de clarificación se incluye el estudio de un ejemplo.

Consulta con elipsis:        Dame las calificaciones de Juan Pérez.

Consulta sin elipsis: Dame las calificaciones de las materias que cursa el alumno Juan Pérez.

Al leer la consulta de ejemplo, una persona sabe que Juan Pérez es un nombre de persona; en cambio una ILNBD no sabe esto, pues sólo le representa un valor de búsqueda. Además, una persona sabe que si se hace la referencia a *calificaciones*, éstas deben corresponder a ciertas materias; por lo tanto, una persona sabe que *Juan Pérez* debe ser el nombre del alumno que cursa las materias cuyas calificaciones se solicitan. Normalmente las ILNBDs no poseen esta información, y en tal caso para que las ILNBDs puedan interpretar fácilmente la consulta debe formularse sin elipsis.

Aunque la consulta de ejemplo (con elipsis) es suficientemente clara para una persona, puede no ser fácil de interpretar para una ILNBD. La razón de esta dificultad es que normalmente las ILNBDs no tienen tanto conocimiento como una persona.

## **2.5 Administrador de diálogo**

El diálogo humano es una actividad conjunta entre dos o más interlocutores y tiene algunas propiedades tales como la existencia de turnos y las implicaciones producto de la conversación. En una ILNBD es recomendable incluir un módulo de diálogo para facilitar aún más al usuario la comunicación con el sistema de información.

## Diálogo

Un diálogo se puede describir como la interacción entre dos partes, en la cual la información es transferida entre las partes mediante un número de turnos secuenciales (un turno se refiere a la transferencia ininterrumpida de información de una parte a la otra) [Niesler, 2001].

## Sistema de diálogo

Un sistema de diálogo es una interfaz hombre máquina, en el cual el usuario puede solicitar información acerca de un dominio específico. La interacción puede ser conducida por la máquina, significando que el sistema guía al usuario, a menudo efectuando preguntas simples y específicas [Rojas, 2009] (Figura 2.3).



Figura 2.3. Arquitectura general de un sistema de diálogo

## Administrador de diálogo

La función de un administrador de diálogo difiere un poco entre las diferentes arquitecturas de sistemas de diálogo, pero su principal responsabilidad es controlar el flujo del diálogo, decidiendo cómo debe responder el sistema a las

expresiones del usuario. Esto se logra inspeccionando y especificando contextualmente la estructura de información producida por un módulo de interpretación. Si la información está incompleta o una solicitud es ambigua, el administrador de diálogo genera preguntas de aclaración al usuario [Kate, 2006].

Un administrador de diálogo puede utilizar un modelo de diálogo, un modelo de tareas y una historia del diálogo [Rohit, 2006] (Figura 2.3):

- a) El modelo de diálogo mantiene una descripción genérica de cómo se construye el diálogo; por ejemplo, para decidir qué acción tomar en una cierta situación. Éste se utiliza para controlar la interacción, lo cual involucra determinar: a) qué es lo que el sistema debe hacer enseguida, y b) decidir qué acción comunicativa es apropiada en un estado de diálogo dado.
- b) El modelo de sistema de tareas define cómo se ejecutan las tareas del sistema.
- c) La historia del diálogo almacena el foco de atención y contiene información acerca de objetos, propiedades y relaciones, así como otra información del diálogo tal como información de actos del habla e información de tareas del sistema.

## **2.6 Antecedentes**

Los trabajos previos que son relevantes para este trabajo de tesis son:

- § Modelo Semánticamente Enriquecido de Bases de Datos para su Explotación por Interfaces de Lenguaje Natural, por Marco Aguirre [Aguirre, 2014], el cual tiene como producto terminado una interfaz de lenguaje natural para

consultas en bases de datos, con la capacidad de traducir consultas de lenguaje natural a SQL. Dicha ILNBD tiene la característica de ser independiente de dominio, pues soporta consultas a cualquier base de datos relacional, debido a su modelo semánticamente enriquecido para bases de datos. Además, permite consultas que incluyan verbos, adjetivos, preposiciones, entre otros, como descriptores de columnas y tablas de una base de datos. La arquitectura de la ILNBD se compone de tres partes generales: el procesamiento de configuración y afinación, el procesamiento de traducción y la administración de diálogo. Esta última parte, corresponde al presente trabajo de tesis.

- § Administrador de Diálogo para una Interfaz de Lenguaje Natural a Bases de Datos, por Carlos Rojas [Rojas, 2009], el cual consiste en el desarrollo de procesos de diálogo independientes del dominio para un administrador de diálogo de una ILNBD desarrollada en el CENIDET. Los procesos de diálogo de este administrador se diseñaron para funcionar de manera independiente del dominio de los datos; es decir, funcionan sobre cualquier base de datos relacional. Esto se logró a través de la implementación de los procesos de diálogo correspondientes a una tipificación formal de problemas en consultas que permite abarcar la mayoría de los casos.

Se considera importante destacar que el administrador de diálogo que se desarrolló en este trabajo, está basado en la tipificación de problemas en consultas de Rojas, pero dirigido a su inclusión en la interfaz de Aguirre, ya que el administrador de diálogo desarrollado por Rojas no es posible acoplarlo a la interfaz desarrollada por Aguirre. La razón esencial de esto es que la interfaz de Aguirre, como se comentó anteriormente, acepta solicitudes de información que contengan sustantivos, verbos, adjetivos y cualquier preposición, considerándolos en la traducción de la consulta a SQL. Esta nueva capacidad requirió el uso de estructuras de datos más complejas a las que se utilizaron en el trabajo de Rojas,

ya que esta interfaz sólo toma en cuenta los sustantivos y la preposición *de*, que contengan las consultas originales de los usuarios.

## 2.7 Trabajos relacionados

Desde los años 60s y hasta la actualidad se han desarrollado muchos trabajos sobre ILNBDs. Sin embargo, todavía existen algunos problemas en ellas que aún no han sido solucionados exitosamente, como la ambigüedad de la interpretación de las palabras, la precisión en las respuestas, la adaptabilidad a nuevos dominios, entre otros [Cimiano, 2007].

En la Tabla 2.1 se describen las interfaces que corresponden a algunos de los trabajos relacionados que se han realizado y que incluyen administradores de diálogo.

Tabla 2.1 Descripción de algunos trabajos relacionados

<b>Interfaz</b>	<b>Descripción</b>
<p>CLARE [Alshawi, 1994]  (1994)</p>	<p>Desarrollado en Cambridge Regional College. Se enfoca a técnicas de procesamiento de lenguaje natural (PLN); utiliza los componentes: dominio léxico, lenguaje de PLN, definición de equivalencias y definición de relaciones funcionales para definir un dominio específico; utiliza un lenguaje de razonamiento (TRL); utiliza diálogos de aclaración como medio de respuesta al usuario.</p>
<p>CoBase [Zhang, 1999]  (1999)</p>	<p>Desarrollado en IBM Santa Tesesa Lab. Se enfoca en el manejo de estructuras de datos; crea tipos abstractos de jerarquías (THAs) para modelar la base de conocimientos; mediante los THAs define operaciones de especialización, generalización y asociación; emplea CoSQL una extensión a SQL; utiliza resultados en SQL como medio de respuesta al usuario. Usa un método incremental que ayuda a los usuarios a resolver consultas complejas formulando una serie de preguntas simples.</p>

Tabla 2.1 Descripción de algunos trabajos relacionados (continuación)

InBase [Boldasov, 2003]  (2003)	Desarrollado en Russian Research Institute. Se enfoca en el manejo de patrones semánticos para consultar la BD; emplea Q-language un lenguaje similar a OQL (Object Query Language); emplea Q-Gen un generador de lenguaje natural; utiliza diálogos de aclaración como medio de respuesta al usuario.
STEP [Minock, 2007]  (2007)	Desarrollado en Umeå University. Se enfoca en un esquema de tuplas de cálculo relacional; utiliza un diccionario de sinónimos de la BD; emplea SPASS un probador de teoremas para consultas; utiliza diálogos de aclaración como medio de respuesta al usuario.
C-PHRASE [Minock, 2010]  (2010)	Desarrollado en Umeå University. Se enfoca en un esquema de alto orden del cálculo de tuplas de Codd; utiliza gramáticas extendidas sincrónicas con funciones lambda para representar las gramáticas semánticas; incluye una evaluación métrica para la precisión; utiliza diálogos de aclaración como medio de respuesta al usuario.

En la Tabla 2.2 se hace una comparación de las ILNBDs de los trabajos relacionados descritos en la Tabla 2.1, de acuerdo a sus características de dominio, diálogos de aclaración y exactitud en las respuestas.

Tabla 2.2 Comparación de las ILNBDs de algunos trabajos relacionados

Características	Interfaces de Lenguaje Natural para Bases de Datos (ILNBDs)					
	Sistema propuesto	(1994) CLARE	(1999) CoBase	(2003) InBase	(2007) STEP	(2010) C-PHRASE
Dominio abierto	Avanzado	Medio	Medio	Medio	Medio	Medio
Diálogos de aclaración	Medio	Medio	Medio	Medio	Medio	Medio
Exactitud en las respuestas	Avanzado	Medio	Medio	Medio	Medio	Medio

Tabla 2.2 Comparación de las ILNBDs de algunos trabajos relacionados  
(continuación)

Simbología	Descripción
Pobre	<p><i>Dominio abierto:</i> se declara como característica, pero las pruebas se han efectuado solamente en un dominio.</p> <p><i>Diálogo de aclaración:</i> éstos se limitan a respuestas sí/no.</p> <p><i>Exactitud en las respuestas:</i> la información retornada al usuario no satisface la pregunta inicial.</p>
Medio	<p><i>Dominio abierto:</i> permite varios dominios, pero la configuración de éstos es en general de forma manual o semiautomática (configuración manual y realizada por el sistema).</p> <p><i>Diálogo de aclaración:</i> no resuelve todos los problemas asociados con sintaxis, semántica, elipsis y anáfora.</p> <p><i>Exactitud en las respuestas:</i> la información retornada al usuario contiene datos de más o de menos, con respecto a la respuesta esperada.</p>
Avanzado	<p><i>Dominio abierto:</i> es aplicable a varios dominios y la configuración es desarrollada en su mayoría por el sistema.</p> <p><i>Diálogo de aclaración:</i> resuelve problemas como sintaxis, semántica, elipsis y anáfora.</p> <p><i>Exactitud en las respuestas:</i> la información retornada al usuario lo conduce a obtener la respuesta esperada.</p>

Cabe señalar que, aunque se han desarrollado muchas ILNBDs, solamente unas pocas incluyen administradores de diálogo. Por ejemplo, aunque en la Tabla 2.1 se mencionan trabajos publicados durante los últimos 15 años, aún así, son escasas las ILNBDs que incluyen diálogo.

Se agregó también el sistema CLARE [Alshawi, 1994], que aunque fue dado a conocer hace casi 20 años, marcó una diferencia con respecto a otras aplicaciones, ya que es un sistema procesador de lenguaje natural, que incluye entre sus características principales, la asistencia al usuario a través de diálogo para la clarificación tanto de consultas como de sentencias de actualización a bases de datos.

Por otro lado, algunas de las interfaces listadas emplean un diálogo de aclaración limitado. Tal es el caso de STEP [Minock, 2007], que ofrece preguntas dicotómicas al usuario para confirmar la consulta, o preguntas de opción múltiple donde le presenta al usuario ejemplos de posibles consultas que podría realizar.

A modo de conclusión, existen muy pocos desarrollos en administración de diálogo para ILNBDs, los cuales no han logrado alcanzar un nivel avanzado de desempeño, por lo que se considera que es un área de oportunidad que se puede abordar en este trabajo.

## **CAPÍTULO 3. METODOLOGÍA**

En este capítulo se presentan las etapas realizadas para el desarrollo del administrador de diálogo. Estas etapas son: el estudio de la tipificación de problemas en consultas; la identificación de los problemas básicos y dobles; el estudio de la estructura de datos que utiliza la ILNBD, el cual es utilizado por el administrador de diálogo de este trabajo; y el tratamiento de problemas para su resolución por medio de diálogo con el usuario.

### **3.1 Problemas en consultas**

Una consulta a base de datos en lenguaje natural puede presentar elipsis por insuficiencia de conocimiento, esto significa que puede haber uno o más elementos de información omitidos, necesarios para la traducción de la consulta de lenguaje natural a SQL. Dependiendo de cuáles piezas de información están ausentes, se reconoce un problema en la consulta.

En la Tabla 3.1 [Rojas, 2009], se presenta la tipificación de problemas en consultas, considerando la relación de tres elementos fundamentales en una consulta a bases de datos: tabla, columna y valor.

Tabla 3.1 Tipificación de problemas considerando los elementos de una consulta a bases de datos

Frase	Elemento			Tipo de problema
	Tabla	Columna	Valor	
Select	ü	?		2.1
	ü	x		6
	?	ü		4.1
	x	ü		7
	?	?		8.1
	x	x		6, 7
Where	ü	?	ü	3.1
	ü	x	ü	6
	?	ü	ü	5.1
	x	ü	ü	7
	?	?	ü	8.2
	x	x	ü	6, 7
	ü, ?, x	ü, ?, x	?	8.3

Notas: El símbolo "ü" significa que el elemento está especificado en la consulta, "?" significa que el elemento no está especificado, y "x" significa que el elemento sí está especificado pero no se encuentra en la base de datos. En el último renglón se excluye la combinación (Tabla, Columna, Valor) = (?, ?, ?), ya que esta combinación implica que no existe frase Where.

Una vez que se estudió la Tabla 3.1, se revisó la tipificación de problemas básicos en consultas (Tabla 3.2), donde se describe cada problemática en términos de los conflictos que aparecen en la consulta con respecto a sus elementos y frases.

### 3.1.1 Problemas básicos en consultas

A continuación se presenta la Tabla 3.2 [Rojas, 2009] que describe cada uno de los problemas que conforman la tipificación de problemas básicos en consultas considerados para el administrador de diálogo.

Tabla 3.2 Tipificación de problemas básicos en consultas

<b>Caso</b>	<b>Descripción</b>
<b>1</b>	<b>Columnas y tablas explícitas</b>
1.1	Consultas que incluyen frases Select y Where y que incluyen explícitamente nombres de columnas y tablas (por lo tanto, no se requiere aclaración de información) <i>Ejemplo: Lista las montañas con elevación mayor a 2000.</i>
1.2	Consultas que incluyen sólo la frase Select y que incluyen explícitamente nombres de columnas y tablas (por lo tanto, no se requiere aclaración de información). <i>Ejemplo: Muéstrame todos los números de vuelo.</i>
<b>2</b>	<b>Columna implícita unívoca en la frase Select</b>
2.1	Consultas que incluyen un nombre de tabla sin especificar cuál(es) de sus columnas se solicita(n). <i>Ejemplo: Lista ¿? los aviones.</i>
<b>3</b>	<b>Columna implícita unívoca en la frase Where</b>
3.1	Consultas que incluyen un nombre de tabla y un valor en la condición de búsqueda sin especificar la columna de tabla relacionada con el valor. <i>Ejemplo: Lista el número de asientos en el avión ¿? M80.</i>
<b>4</b>	<b>Columna implícita multívoca en la frase Select</b>
4.1	Consultas que incluyen el nombre de una columna sin especificar una de las varias tablas a las que puede estar relacionada. <i>Ejemplo: Muestra los teléfonos.</i> <i>Nota: Existen tres tablas con la columna teléfono.</i>
<b>5</b>	<b>Columna implícita multívoca en la frase Where</b>
5.1	Consultas que incluyen el nombre de una columna sin especificar una de las varias tablas a las que puede estar relacionada. <i>Ejemplo: Lista la categoría de aeroplano para el código WLH.</i> <i>Nota: Existen dos tablas con la columna código.</i>
<b>6</b>	<b>Columna inexistente</b>
6.1	Consultas que incluyen un término que podría referirse al nombre de una columna que no puede ser relacionado a ninguna tabla de la base de datos. <i>Ejemplo: Muestra los tamaños de los aviones.</i> <i>Nota: No existe la columna tamaño en la BD.</i>

Tabla 3.2 Tipificación de problemas básicos en consultas (continuación)

7	<b>Tabla inexistente</b>
7.1	Consultas que incluyen un término que podría referirse al nombre de una tabla que no pertenece a la base de datos. <i>Ejemplo: Muestra las áreas de los condados.</i> <i>Nota: No existe la tabla condados en la BD.</i>
8	<b>Consultas mal formadas</b>
8.1	Consultas en las que en la frase Select no se especifica ni el nombre de una columna ni el nombre de una tabla. <i>Ejemplo: Muéstralos todos.</i>
8.2	Consultas que incluyen en la frase Where un valor en la condición de búsqueda sin especificar ni el nombre de una columna ni el nombre de una tabla. <i>Ejemplo: Muéstrame el ¿? D10.</i>
8.3	Consultas que incluyen en la frase Where el nombre de una columna y/o el nombre de una tabla sin especificar un valor en la condición de búsqueda. <i>Ejemplo: Lista las órdenes enviadas al cliente ¿?.</i>

Los problemas cuyo tratamiento no se incluyeron en el administrador de diálogo son los que corresponden a los casos 1, 6, 7 y 8; por lo tanto, los problemas que sí tienen tratamiento en el administrador de diálogo de este trabajo de tesis son los casos específicos designados como 2.1, 3.1, 4.1 y 5.1.

### 3.1.2 Problemas dobles en consultas

A partir de los casos de problemas considerados para el administrador de diálogo, se elaboró una tipificación de problemas dobles (Tabla 3.3) con las posibles combinaciones de dos problemas diferentes de los cuatro problemas individuales. Esta tipificación se desarrolló con el propósito de estudiar el tratamiento que se debía dar a estos casos, pues en algunos se presenta un problema en la frase Select y otro en la frase Where. La idea fue determinar cuál problema debía tratarse primero para intentar una implementación más eficaz para el tratamiento de estos casos con problemas dobles.

Tabla 3.3 Tipificación de problemas dobles

Caso	Descripción
a	<p><b>Problemas 2.1 y 3.1</b></p> <p>La consulta incluye en la frase Select un nombre de tabla sin especificar algún nombre de columna; además, incluye en la frase Where un nombre de tabla y un valor de búsqueda sin especificar el nombre de columna al que corresponde.</p> <p><i>Ejemplo: Muéstrame la tarifa para el vuelo 106245.</i></p> <p><i>Nota: No se especifican columnas para la tabla tarifa y no se especifica la columna de la tabla vuelo.</i></p>
b	<p><b>Problemas 2.1 y 4.1</b></p> <p>La consulta incluye en la frase Select un nombre de tabla sin especificar algún nombre de columna correspondiente; además, incluye un nombre de columna sin especificar una de las varias tablas a las que puede corresponder.</p> <p><i>Ejemplo: Lista precios y categorías.</i></p> <p><i>Nota: No se especifican columnas para la tabla categorías y existen dos tablas con la columna precio.</i></p>
c	<p><b>Problemas 2.1 y 5.1</b></p> <p>La consulta incluye en la frase Select un nombre de tabla sin especificar algún nombre de columna; además, incluye en la frase Where un valor de búsqueda y un nombre de columna sin especificar una de las varias tablas a las que puede corresponder.</p> <p><i>Ejemplo: ¿Cuáles órdenes tienen descuentos del 0.5?</i></p> <p><i>Nota: No se especifican columnas para la tabla órdenes y existen dos tablas con la columna descuentos.</i></p>
d	<p><b>Problemas 4.1 y 3.1</b></p> <p>La consulta incluye en la frase Select un nombre de columna sin especificar una de las varias tablas a las que puede corresponder; además, incluye en la frase Where un nombre de tabla y un valor de búsqueda sin especificar el nombre de columna al que corresponde.</p> <p><i>Ejemplo: ¿Cuál es el número de escalas en el vuelo 21?</i></p> <p><i>Nota: No se especifica la columna de la tabla vuelo y existen dos tablas con la columna número de escalas.</i></p>
e	<p><b>Problemas 4.1 y 5.1</b></p> <p>La consulta incluye en la frase Select un nombre de columna sin especificar una de las varias tablas a las que puede corresponder; además, incluye en la frase Where un valor de búsqueda y un nombre de columna sin especificar una de las varias tablas a las que puede corresponder.</p> <p><i>Ejemplo: Lista las poblaciones de las áreas menores a 5000.</i></p> <p><i>Nota: Existen dos tablas con la columna población y existen dos tablas con la columna área.</i></p>

Tabla 3.3 Tipificación de problemas dobles (continuación)

f	<p><b>Problemas 3.1 y 5.1</b></p> <p>La consulta incluye en la frase Where un nombre de tabla y un valor de búsqueda sin especificar el nombre de columna al que corresponde; además, incluye un valor de búsqueda y un nombre de columna sin especificar una de las varias tablas a las que puede corresponder.</p> <p><i>Ejemplo: ¿Cuál es la tarifa redonda para los vuelos US con un tiempo de viaje menor a 50?</i></p> <p><i>Nota: No se especifica la columna de la tabla vuelo y existen dos tablas con la columna tiempo de viaje.</i></p>
---	---

Cabe señalar que las descripciones redactadas para esta tipificación se basan en las descripciones de la tipificación de problemas individuales, adaptándolas a la situación conjunta de dos problemas. Además, no se consideraron los casos de las consultas con dos problemas del mismo tipo individual, es decir, el caso de una consulta con dos problemas tipo 2.1, por mencionar un ejemplo; ya que el orden del tratamiento, para un caso con dos problemas del mismo tipo, no influye en la eficacia de la resolución del caso.

### 3.2 Estructura de datos

El administrador de diálogo de este trabajo de tesis, se acopló a la ILNBD desarrollada en el ITCM [Aguirre, 2014], por lo que fue necesario estudiar la estructura de datos de dicha interfaz, para identificar los casos de los problemas en las consultas, así como para realizar el llenado de los datos necesarios para la correcta traducción de la consulta en lenguaje natural a una consulta en SQL.

La ILNBD fue programada en Java, el cual es un lenguaje de programación orientado a objetos; por lo mismo, la estructura de datos que utiliza para tratar la consulta en lenguaje natural, está implementada por medio de clases, con atributos y métodos. En esta estructura se organiza la consulta en lenguaje natural

con los datos resultantes de la primera fase del proceso de traducción; es decir, se desglosa la consulta en componentes léxicos y los etiqueta de acuerdo a los análisis léxico, sintáctico y semántico. En la segunda fase, se elabora una consulta en SQL en base a la información de la estructura de datos mencionada. El administrador de diálogo actúa sobre esta estructura entre la primera fase y la segunda.

Para una mejor comprensión, se representa esta estructura en forma de tabla (Tabla 3.4), donde sólo se incluyen los atributos que se necesitan para la identificación de problemas en la consulta y los atributos que se requieren llenar con los datos recolectados a partir del diálogo con el usuario. Este llenado de datos es el resultado del tratamiento que hace el administrador de diálogo para aclarar la consulta, y de esta manera, completar los datos que se ocupan para la traducción de la consulta en lenguaje natural a su correspondiente consulta en SQL.

En la Tabla 3.4, se muestra la representación de la estructura de datos de la ILNBD con una consulta de ejemplo procesada. La consulta de ejemplo es: *lista los aviones con velocidad mayor a 400*.

Tabla 3.4 Representación de la estructura de datos de la ILNBD con una consulta procesada

Atributo	Unidades léxicas							
Componente léxico	<i>lista</i>	<i>los</i>	<i>aviones</i>	<i>con</i>	<i>velocidad</i>	<i>mayor</i>	<i>a</i>	<i>400</i>
Categoría gramatical	orden	artículo	sustantivo	preposición	sustantivo	comparación	preposición	entero
Frase			avión		velocidad			
Tipo de frase			Select		Where			Where
Etiqueta de columna					aircraft. cruising.speed			
Etiqueta de tabla			aircraft					
Etiqueta final			aircraft		aircraft. cruising.speed	>		400
Tabla	false	false	true	false	false	false	false	false
Columna	false	false	false	false	true	false	false	false
Marcado	false	false	true	false	true	true	false	true

Los atributos de la estructura de datos que son relevantes para este trabajo son:

- § **Componente léxico:** Contiene una unidad léxica del texto en la consulta original en lenguaje natural.
- § **Categoría gramatical:** Contiene la clase gramatical a la que corresponde la unidad léxica (por ejemplo: sustantivo, verbo, adjetivo, entero, etc.). Para el administrador de diálogo es útil conocer el tipo de valor de búsqueda, y en este atributo se almacena esta información para esa clase de unidades léxicas.
- § **Frase:** Contiene la frase o sintagma (construido por unidades léxicas) con la raíz de los componentes léxicos, que se relaciona con las descripciones de los elementos (tabla o columna) en la base de datos que se esté consultando. Esta información le sirve al administrador de diálogo para realizar búsquedas de tablas o columnas referenciadas en la consulta.
- § **Tipo de frase:** Especifica qué tipo de frase es: Select o Where. El administrador de diálogo utiliza esta información para realizar el proceso de identificación de problemas.
- § **Etiqueta de columna:** Contiene un listado con los nombres de columna de la base de datos que se esté consultando, que corresponden a la frase de la unidad léxica. El contenido de este atributo, permite al administrador de diálogo verificar cuántas referencias existen hacia la columna especificada en la consulta.
- § **Etiqueta de tabla:** Contiene un listado con los nombres de tabla de la base de datos que se esté consultando, que corresponden a la frase de la unidad léxica. El administrador de diálogo utiliza esta información para realizar una búsqueda de las columnas de la tabla referenciada en la consulta.
- § **Etiqueta final:** Contiene el nombre de la columna o la tabla de la base de datos que se esté consultando, que corresponde a la frase de la unidad

léxica. En este atributo, el administrador de diálogo, realiza modificaciones en caso de ser necesario, según el proceso de diálogo que realice.

§ **Tabla:** Es un valor lógico que indica si la unidad léxica corresponde a una tabla de la base de datos que se esté consultando. El administrador de diálogo utiliza este valor en el proceso de identificación de problemas que incluyen nombres de tablas.

§ **Columna:** Es un valor lógico que indica si la unidad léxica corresponde a una columna de la base de datos que se esté consultando. El administrador de diálogo utiliza este valor en el proceso de identificación de problemas que incluyen nombres de columnas.

§ **Marcado:** Es un valor lógico que indica si la unidad léxica será un elemento necesario para la traducción a SQL. El administrador de diálogo verifica que la unidad léxica se debe revisar si este atributo tiene un valor de verdadero.

### 3.3 Algoritmos

Los algoritmos que se diseñaron para este trabajo de tesis son: el algoritmo de identificación de problemas y los algoritmos de los procesos de diálogo.

#### 3.3.1 Algoritmos de identificación de problemas

El algoritmo de identificación de existencia de problemas en la consulta (ver Algoritmo 3.1), detecta si existe algún caso de la tipificación de problemas individuales, y al mismo tiempo, revisa si se presenta algún caso con dos problemas.

La revisión de la consulta es efectuada, desde un algoritmo principal, en dos fases: la primera para la frase Where de la consulta (ver Algoritmo 3.2), y la segunda para la frase Select (ver Algoritmo 3.3).

El algoritmo funciona con la estructura de datos de la ILNBD comentada en la sección 3.2, utilizando los valores de los atributos de dicha estructura para verificar si se presenta alguno de los casos de problemas.

---

*Algoritmo 3.1 Pseudocódigo de identificación de existencia de problemas*

---

```
1   E // Estructura de datos de la ILNBD
2   if algunComponenteLexicoSinCategoríaGramatical(E)
3       problema6o7()
4   else
5       if existeValorBusqueda(E) // Verifica que exista frase Where
6           revisionFraseWhere(E) // Algoritmo 3.2
7       else
8           if existeColumnaOTablaEnFraseWhere(E)
9               problema8.3()
10          endif
11      endif
12      revisionFraseSelect(E) // Algoritmo 3.3
13      if algunProblemaATratarConAdministradorDeDialogo()
14          activarDialogoParaProblema()
15      else
16          consultaSinElipsis()
17      endif
18  endif
```

---

---

*Algoritmo 3.2 Pseudocódigo de revisión en frase Where*

---

```
1   n // Número total de unidades léxicas en E
2   for i = 0, ..., n-1 do
3       if esColumnaEnFraseWhere(Ei)
4           if |etiquetaColumna(Ei)| > 1
5               problema5.1()
6           endif
7       endif
8       if esTablaEnFraseWhere(Ei)
```

```

9           if noExisteColumnaEnFraseWhereDeTablaEi(E)
10              problema3.1()
11         endif
12     endif
13 endfor
14 if noExisteColumnaNiTablaEnFraseWhere(E)
15     problema8.2()
16 endif

```

---



---

*Algoritmo 3.3 Pseudocódigo de revisión en frase Select*

---

```

1   for i = 0, ..., n-1 do
2       if esColumnaEnFraseSelect(Ei)
3           if |etiquetaColumna(Ei)| > 1
4               problema4.1()
5           endif
6       endif
7       if esTablaEnFraseSelect(Ei)
8           if noExisteColumnaEnFraseSelectDeTablaEi(E)
9               problema2.1()
10          endif
11      endif
12  endfor
13  if noExisteColumnaNiTablaEnFraseSelect(E)
14      problema8.1()
15  endif

```

---

Estos algoritmos se apegan a las descripciones de los casos en la tipificación de problemas. Para una mejor comprensión del algoritmo de identificación de problemas, se incluye un ejemplo de cómo se lleva a cabo la identificación del problema 2.1 en la consulta: *lista los aviones con velocidad mayor a 400*, utilizando la estructura de datos con la consulta procesada (Tabla 3.5) y siguiendo la lógica del algoritmo:

1. Inicia con la revisión de que todas las unidades léxicas tengan una categoría gramatical (ver Algoritmo 3.1 línea 2). En este ejemplo, la

condición se cumple, ya que se observa que todas las unidades léxicas poseen una categoría gramatical.

2. Verifica la existencia de un valor de búsqueda (ver Algoritmo 3.1 línea 5). En este ejemplo, la condición se cumple, pues la última unidad léxica de la consulta contiene el texto “valor” en el atributo *Categoría gramatical*.
3. Continúa con la revisión de la frase Where (ver Algoritmo 3.1 línea 6).
4. Verifica si existe la referencia a una columna con tipo de frase Where (ver Algoritmo 3.2 línea 3). En este ejemplo, se observa que existe una unidad léxica con los datos: *Tipo de frase*(“Where”), *Columna*(true) y *Marcado*(true). Por lo que se cumple la condición.
5. Verifica si el elemento tiene más de una etiqueta de columna (ver Algoritmo 3.2 línea 4). En este ejemplo, se observa que el atributo *Etiqueta de columna* contiene sólo una referencia de columna. Por lo tanto, no se cumple la condición.
6. Verifica si existe la referencia a una tabla con tipo de frase Where (ver Algoritmo 3.2 línea 8). En este ejemplo, se observa que no existe una unidad léxica que cumpla la condición.
7. Revisa que no exista referencia a una columna o una tabla con tipo de frase Where (ver Algoritmo 3.2 línea 14). En este ejemplo, en el punto 4 se verificó la existencia de una referencia a columna, entonces, no se cumple la condición.
8. Continúa con la revisión de la frase Select (ver Algoritmo 3.1 línea 12).

9. Verifica si existe la referencia a una columna con tipo de frase Select (ver Algoritmo 3.3 línea 2). En este ejemplo, se observa que no existe una unidad léxica que cumpla la condición.
10. Verifica si existe la referencia a una tabla con tipo de frase Select (ver Algoritmo 3.3 línea 7). En este ejemplo, se observa que existe una unidad léxica con los datos: *Tipo de frase*("Select"), *Tabla*(true) y *Marcado*(true). Por lo que se cumple la condición.
11. Revisa si no existe la referencia a una columna con tipo de frase Select que corresponda a la tabla verificada (ver Algoritmo 3.3 línea 8). En este ejemplo, se observa que no existe una unidad léxica con los datos: *Tipo de frase*("Select"), *Columna*(true) y *Marcado*(true). Por lo que se cumple la condición.
12. Identifica la presencia del problema 2.1 (ver Algoritmo 3.3 línea 9).

Tabla 3.5 Representación de la estructura de datos de la ILNBD con una consulta procesada de ejemplo con el problema 2.1

Atributo	Unidades léxicas							
	<i>lista</i>	<i>los</i>	<i>aviones</i>	<i>con</i>	<i>velocidad</i>	<i>mayor</i>	<i>a</i>	<i>400</i>
Componente léxico	<i>lista</i>	<i>los</i>	<i>aviones</i>	<i>con</i>	<i>velocidad</i>	<i>mayor</i>	<i>a</i>	<i>400</i>
Categoría gramatical	orden	artículo	sustantivo	preposición	sustantivo	comparación	preposición	entero
Frase			avión		velocidad			
Tipo de frase			Select		Where			Where
Etiqueta de columna					aircraft. cruising.speed			
Etiqueta de tabla			aircraft					
Etiqueta final			aircraft		aircraft. cruising.speed	>		400
Tabla	false	false	true	false	false	false	false	false
Columna	false	false	false	false	true	false	false	false
Marcado	false	false	true	false	true	true	false	true

También es posible comprobar que la consulta de ejemplo involucra el problema 2.1, parafraseando la descripción de este problema: *Consultas que incluyen un nombre de tabla, que corresponda a la frase Select, sin especificar cuál(es) de sus columnas se solicita(n)*, y aplicándola a la estructura de datos en la Tabla 3.5, se observa que se incluye un nombre de tabla (*aircraft*) identificada en la frase Select (en la unidad léxica sombreada en la tabla), y también se observa, que aunque existe una especificación de columna de esa tabla, no corresponde a la frase Select. Por lo tanto, se concluye que la consulta de ejemplo presenta el problema 2.1.

### 3.3.2 Algoritmos de los procesos de diálogo

Los procesos de diálogo inician después de la identificación de los problemas a tratar, dependiendo del problema es el proceso de diálogo que se ejecuta.

A continuación se presentan los procesos de diálogo diseñados para el tratamiento de los problemas.

---

#### Algoritmo 3.4 Pseudocódigo del proceso de diálogo para el problema 2.1

---

```
1   T // Tabla
2   L,S // Listas temporales
3   m // Número de elementos en S
4   T! localizarTablaConProblema2.1()
5   L! buscarColumnas(T)
6   desplegarColumnas(L)
7   solicitarSeleccionDeColumnas()
8   S! columnasSeleccionadasPorUsuario()
9   for i = 0,..., m-1 do
10      E! agregarUnidadLexica(Si)
11  endfor
```

---

---

*Algoritmo 3.5 Pseudocódigo del proceso de diálogo para el problema 3.1*

---

```
1   C // Columna
2   j // Posición de la unidad léxica con el problema 3.1
3   L // Lista temporal
4   T! localizarTablaConProblema3.1()
5   L! buscarColumnas(T)
6   desplegarColumnas(L)
7   solicitarSeleccionDeLaColumna() // correspondiente al valor de búsqueda
8   C! columnaSeleccionadaPorUsuario()
9   modificarAtributosDeUnidadLexica(Ej, C)
```

---

---

*Algoritmo 3.6 Pseudocódigo del proceso de diálogo para los problemas 3.1 y 5.1*

---

```
1   k // Posición de la unidad léxica con el problema 4.1 o 5.1
2   C! localizarColumnaConProblema3.1-5.1()
3   L! buscarTablasConLaColumna(C)
4   desplegarTablas(L)
5   solicitarSeleccionDeLaTabla() // correspondiente a la columna
6   T! tablaSeleccionadaPorUsuario()
7   modificarUnidadLexica(Ek, T)
```

---

Cabe mencionar que el proceso de diálogo para los problemas 4.1 y 5.1 es el mismo, pues el enfoque de resolución para la problemática donde se desconoce a cuál tabla corresponde una columna especificada en la consulta, es similar ya sea en la frase *Select* o en la frase *Where*.

En seguida, se muestra un ejemplo de cómo interactúa el administrador de diálogo con el usuario y con la estructura de datos de la ILNBD. La consulta de ejemplo: *lista las poblaciones*, tiene el problema 4.1, es decir, se especifica en la consulta la referencia a un nombre de columna, pero no se indica a cuál tabla pertenece, perteneciendo la columna a dos tablas. En la Tabla 3.6 se representa el contenido de la estructura de datos de la ILNBD con la consulta procesada.

Tabla 3.6 Representación de la estructura de datos de la ILNBD con una consulta procesada de ejemplo con el problema 4.1

Atributo	Unidades léxicas		
Componente léxico	<i>lista</i>	<i>las</i>	<i>poblaciones</i>
Categoría gramatical	orden	artículo	sustantivo
Frase			población
Tipo de frase			Select
Etiqueta de columna			state.population city.population
Etiqueta final			
Columna	false	false	true
Marcado	false	false	true

Como se observa en la Tabla 3.6, la unidad léxica sombreada es la que concierne a la especificación de columna, y en el atributo *Etiqueta de columna* se aprecia que contiene dos etiquetas de columna; es decir, la unidad léxica hace referencia a una columna que aparece en dos tablas, por lo que presenta el problema 4.1.

Al activarse el administrador de diálogo, ejecuta el proceso de diálogo para el problema en cuestión:

1. Identifica la columna especificada en la consulta (ver Algoritmo 3.6 línea 2). En este caso, la columna con el conflicto es *población (population)*.
2. Realiza la búsqueda de las tablas que contengan la columna (ver Algoritmo 3.6 línea 3), es decir, busca en la base de datos las descripciones de las tablas que contengan la columna en cuestión, para este caso: *estado (state)* y *ciudad (city)*.
3. Presenta las tablas localizadas al usuario (ver Algoritmo 3.6 línea 4). En una interfaz muestra al usuario sólo las descripciones de tabla: *estado* y *ciudad*.

4. Solicita al usuario que indique la tabla correspondiente a la columna especificada (ver Algoritmo 3.6 línea 5), permitiendo al usuario realizar la selección de sólo una de las opciones listadas, ya sea *estado* o *ciudad*.
5. Modifica la estructura de datos de la consulta en la unidad léxica correspondiente a la columna especificada, con los valores de atributos adecuados (ver Algoritmo 3.6 línea 7). Una vez que el usuario seleccionó la opción deseada (para este ejemplo se supondrá que escogió *estado*), se modifica la estructura de datos de la consulta de manera que los datos estén completos, como en la Tabla 3.7.

Tabla 3.7 Representación de la estructura de datos de la ILNBD con una consulta de ejemplo sin elipsis por insuficiencia de conocimiento

Atributo	Unidades léxicas		
Componente léxico	<i>lista</i>	<i>las</i>	<i>poblaciones</i>
Categoría gramatical	orden	artículo	sustantivo
Frase			población
Tipo de frase			Select
Etiqueta de columna			state.population
Etiqueta de tabla			
Etiqueta final			state.population
Tabla	false	false	false
Columna	false	false	true
Marcado	false	false	true

En la Tabla 3.7 se sombrearon los valores de los atributos modificados por el administrador de diálogo, con el propósito de que los datos en la estructura de datos de la consulta procesada queden completos para su traducción a SQL.

Por último, se considera importante señalar que estos procesos de diálogo tratan un problema individual a la vez, así que en los casos con problemas dobles, se ejecutan dos procesos de diálogo dependiendo del par de problemas. Sin embargo, si el conjunto de problemas incluye una problemática en la frase Select y

otra en la frase Where, primero se ejecuta el proceso de diálogo para resolver el problema de la frase Where y después el problema de la frase Select. Esto se debió a que en las pruebas preliminares del funcionamiento de la codificación de los procesos de diálogo, con algunas consultas que involucraban casos con problemas dobles, se observó que al resolverse la problemática en la frase Where, se resolvía el problema en la frase Select, mientras que en otras consultas era indiferente cuál frase se resolvía primero. Por lo tanto, se tomó la decisión de que en consultas con problemas dobles, el orden de ejecución de los procesos de diálogo para los dos problemas sería: primero tratar el problema en la frase Where y posteriormente el de la frase Select.

Un ejemplo de uno de los casos donde resulta exitoso aplicar esta política, se explica a continuación:

En la Figura 3.1 se muestran tres tablas de una base de datos (Proveedores, Clientes y Empleados), donde cada una tiene dos columnas con la misma descripción de dato (Ciudad y Region).



Figura 3.1 Tablas de una base de datos de ejemplo

En una consulta de ejemplo: *¿Cuáles son las ciudades de la región WA?*, se muestran los elementos de la consulta en la Tabla 3.8, donde se aprecia que

existen dos problemas en la consulta, uno en la frase *Select* y otro en la frase *Where*. Los problemas en cuestión se refieren a la especificación de una columna sin indicar a cuál tabla pertenece, cuando ocurre en la frase *Select*, es el problema 4.1, y cuando ocurre en la frase *Where*, es el problema 5.1.

Tabla 3.8 Elementos de una consulta de ejemplo

Frase <i>Select</i>		Frase <i>Where</i>		
Tabla	Columna	Tabla	Columna	Valor
?	proveedores.ciudad clientes.ciudad empleados.ciudad	?	proveedores.region clientes.region empleados.region	WA

Cuando se resuelve el problema en la frase *Where* a través del proceso de diálogo correspondiente, entonces queda resuelto el problema en la frase *Select*, sin requerir otro ciclo de diálogo. Ya que al indicar a cuál tabla pertenece la columna especificada como *region*, se determina que la columna *ciudad* pertenece a la misma tabla, para construir una consulta lógica.

### 3.4 Interfaces de usuario

Para las pantallas del administrador de diálogo, se emplearon objetos visuales de la biblioteca gráfica estándar de Java, para permitir una presentación adecuada de la información al usuario, así como para facilitar el ingreso de respuestas por parte del usuario durante el diálogo. En seguida se presentan las pantallas del administrador de diálogo para cada problema que requiere tratamiento.

La Figura 3.2 es la pantalla para el diálogo del problema 2.1, en ésta se observa que los elementos visuales que se utilizaron para que el usuario responda a la solicitud del administrador de diálogo son básicamente dos listas y dos botones, en la lista de la izquierda se le presentan las columnas que puede seleccionar de la tabla especificada en la consulta, en la segunda lista, el

administrador despliega las columnas que el usuario ha seleccionado al presionar el botón superior (>>) una vez que seleccionó la(s) columna(s) deseada(s) de la primera lista. Se incluyó el botón inferior (<<) para que el usuario retire alguna(s) columna(s) ya seleccionada en la lista de la derecha.

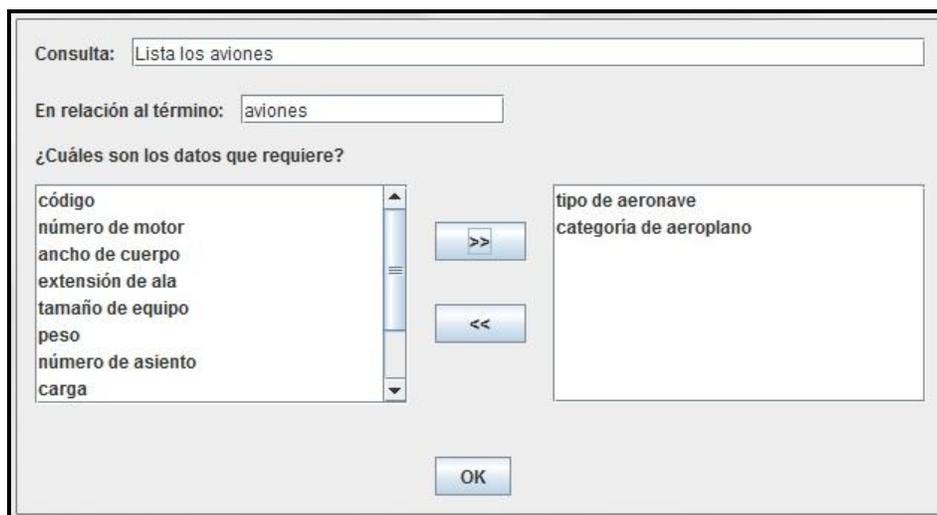


Figura 3.2 Pantalla para el diálogo del problema 2.1

En la Figura 3.3 se muestra la pantalla para el diálogo del problema 3.1. En esta pantalla sólo se agregó una lista para que el usuario seleccione únicamente una de las opciones listadas, a modo de respuesta a la solicitud del administrador de diálogo.

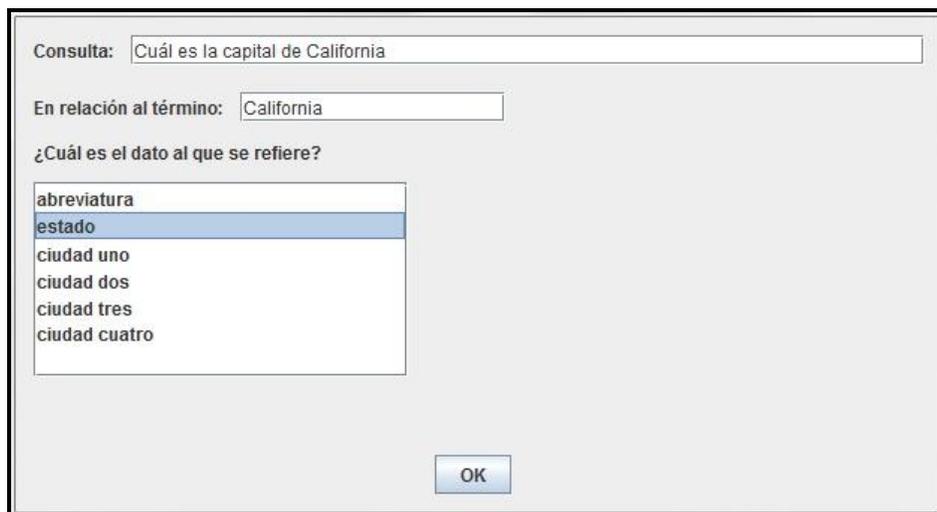


Figura 3.3 Pantalla para el diálogo del problema 3.1

La pantalla para el diálogo de los problemas 4.1 y 5.1 que se muestra en la Figura 3.4, incluye un listado que permite sólo una selección de las opciones listadas por parte del usuario para responder a la solicitud del administrador de diálogo.



Consulta:

En relación al término:

¿A qué se refiere?

código de aeropuerto de [aeropuerto]  
 código de aeropuerto de [servicio de aeropuerto]  
 código de aeropuerto de [servicios de transporte terrestre]

Figura 3.4 Pantalla para el diálogo de los problemas 4.1 y 5.1

## CAPÍTULO 4. EXPERIMENTACIÓN

En este cuarto capítulo se describen los elementos y las pruebas de la experimentación efectuada en el presente trabajo, así como los resultados obtenidos.

### 4.1 Corpus de consultas para prueba

Para la realización de las pruebas del administrador de diálogo se organizó un corpus de consultas a las bases de datos ATIS [UPenn, 1990], Geobase [UTexas, 1995] y Northwind [Bhootra, 2004] [Nihalani, 2011]. Este corpus se integró con 10 consultas para cada uno de los cuatro problemas individuales de la tipificación considerada para el administrador de diálogo, así como 10 consultas para cada uno de los seis casos de la tipificación de problemas dobles, dando un total de 100 consultas, las cuales se detallan en el Anexo A.

La composición del corpus de acuerdo a la base de datos que concierne se presenta en la Tabla 4.1, mientras que la Tabla 4.2 muestra la composición del corpus según los casos de prueba.

Tabla 4.1 Composición del corpus de consultas de prueba según las bases de datos

<b>Base de datos</b>	<b>Número de consultas</b>
ATIS	55
Geobase	15
Northwind	30
<i>Total</i>	100

Tabla 4.2 Composición del corpus de consultas de prueba según los casos de prueba

<b>Caso de prueba</b>	<b>Número de consultas</b>
Problema 2.1	$4 + 6^* = 10$
Problema 3.1	$10 + 0^* = 10$
Problema 4.1	$3 + 7^* = 10$
Problema 5.1	$3 + 7^* = 10$
Problema 2.1 y 3.1	$4 + 6^* = 10$
Problema 2.1 y 4.1	$2 + 8^* = 10$
Problema 2.1 y 5.1	$3 + 7^* = 10$
Problema 4.1 y 3.1	$7 + 3^* = 10$
Problema 4.1 y 5.1	$4 + 6^* = 10$
Problema 3.1 y 5.1	$1 + 9^* = 10$
<i>Total</i>	100
* Consultas propuestas	

La nota incluida en la Tabla 4.2 se refiere a la cantidad de consultas que se propusieron para el corpus de prueba, ya que los corpus existentes de las bases de datos utilizadas no contaban con el número requerido de consultas que se clasifican en los casos de problemas tratados por el administrador de diálogo.

## 4.2 Pruebas

Las pruebas que se realizaron para este trabajo fueron dos: una prueba de desempeño y una prueba de comparación. En las secciones siguientes se describen cada una de ellas.

Ambas pruebas se efectuaron en una computadora de escritorio con las características especificadas en la Tabla 4.3.

Tabla 4.3 Características del equipo

<b>Característica</b>	<b>Especificación</b>
Procesador	Intel Pentium @ 2.26 GHz
Memoria	1 GB
Sistema operativo	Microsoft Windows XP Profesional SP3

#### 4.2.1 Prueba de desempeño

La prueba de desempeño que se realizó para este trabajo, consistió en el estudio de las respuestas que proporcionó la ILNBD con y sin el apoyo del administrador de diálogo, a las consultas en lenguaje natural del corpus mencionado previamente.

En el estudio, se revisó que se le proporcionara al usuario una respuesta a la consulta ingresada, y además, que dicha respuesta fuese exactamente la información solicitada, lo que se consideró como resultado correcto.

En esta experimentación se probaron las 100 consultas del corpus de prueba (ver Anexo A).

Para esta prueba de desempeño, se empleó el mismo software en el que se desarrolló al administrador de diálogo, y se detalla en la Tabla 4.4,

Tabla 4.4 Características del software

<b>Característica</b>	<b>Especificación</b>
Entorno	NetBeans IDE 7.4
Lenguaje	Java 1.7

## 4.2.2 Prueba comparativa

Este trabajo de tesis tiene como antecedente la ILNBD desarrollada en CENIDET [Rojas, 2009], la cual cuenta con un administrador de diálogo basado en una tipificación de problemas en consultas, misma que se aplicó en el presente trabajo. Por tal motivo, se consideró interesante comparar ambas versiones del administrador de diálogo. Es conveniente aclarar que el administrador de diálogo de este trabajo se adaptó a la ILNBD desarrollada en el ITCM [Aguirre, 2014], por lo que los núcleos de las interfaces son diferentes, como se planteó en la sección 2.6.

Esta prueba de comparación cotejó los resultados a 85 consultas del corpus de prueba propuesto para la experimentación, específicamente las consultas a las bases de datos ATIS y Northwind. Se usaron las mismas bases de datos con las mismas descripciones de tablas y columnas, para comparar los datos de respuesta a cada consulta ingresada.

En la comparación, se revisó que el usuario obtuviera un resultado para la consulta y que la información proporcionada por cada interfaz fuese exactamente la solicitada (ni de más ni de menos), además, se comparó la cantidad de ciclos de diálogo con el usuario para la clarificación de cada consulta.

El software necesario para ejecutar la prueba fue diferente para cada interfaz debido a la implementación de cada una. En la Tabla 4.5 se indica el software para cada interfaz.

Tabla 4.5 Software utilizado en la prueba de comparación entre ILNBDs

<b>Software</b>	<b>ILNBD (CENIDET)</b>	<b>ILNBD (ITCM)</b>
Servidor	Apache Tomcat 6.0.13	No aplica
Manejador de BD	Microsoft Access 2007	Microsoft Access 2007
Plataforma	JDK de Java 1.7	JDK de Java 1.7
Entorno	Mozilla Firefox 30.0	NetBeans IDE 7.4

### 4.3 Resultados

Los resultados obtenidos de esta experimentación se describen por separado, para cada una de las dos pruebas.

En la Tabla 4.6 se muestran los resultados de la prueba de desempeño, aclarando que *Provee resultado* significa que el usuario obtiene una respuesta a la consulta ingresada, mientras que *Resultado correcto* significa que los datos obtenidos a partir de la consulta son exactamente los solicitados (ni más ni menos).

Tabla 4.6 Resultados de la prueba de desempeño

	<b>Sin administrador de diálogo</b>		<b>Con administrador de diálogo</b>	
Consultas	100	100%	100	100%
Provee resultado	10	10%	100	100%
Resultado correcto	10	10%	100	100%

En lo que respecta a la prueba de comparación entre las ILNBDs con administrador de diálogo, los resultados en cuanto a los rubros *Provee resultado* y *Resultado correcto* están registrados en la Tabla 4.7 y la Tabla 4.8. En cuanto a los ciclos de diálogo, los resultados se muestran en la Tabla 4.9, considerando sólo las consultas que arrojaron resultado correcto.

Tabla 4.7 Resultados resumidos de la prueba de comparación

	<b>ILNBD [CENIDET]</b>		<b>ILNBD [ITCM]</b>	
Consultas	85	100%	85	100%
Provee resultado	30	35%	85	100%
Resultado correcto	25	29%	85	100%

Tabla 4.8 Resultados de la prueba de comparación de la ILNBD (CENIDET)

Caso	Consultas	Provee resultado		Resultado correcto	
Problema 2.1	10	7	70%	6	60%
Problema 3.1	5	3	60%	3	60%
Problema 4.1	7	4	57%	3	43%
Problema 5.1	7	4	57%	3	43%
Problemas 2.1 y 3.1	10	6	60%	6	60%
Problemas 2.1 y 4.1	10	0	0%	0	0%
Problemas 2.1 y 5.1	10	3	30%	2	20%
Problemas 4.1 y 3.1	8	2	25%	1	13%
Problemas 4.1 y 5.1	8	1	13%	1	13%
Problemas 3.1 y 5.1	10	0	0%	0	0%
<i>Total</i>	85	30	35%	25	29%

Tabla 4.9 Resultados de la prueba de comparación del número de ciclos de diálogo

Caso	Consultas	Promedio de Ciclos	
		ILNBD (CENIDET)	ILNBD (ITCM)
Problema 2.1	6	1	1
Problema 3.1	3	2	1
Problema 4.1	3	1	1
Problema 5.1	3	1	1
Problemas 2.1 y 3.1	6	2	2
Problemas 2.1 y 5.1	2	3	1.5
Problemas 4.1 y 3.1	1	2	2
Problemas 4.1 y 5.1	1	1	1
<i>Total</i>	25	1.63	1.31

En la Tabla 4.9, se tomaron en cuenta sólo las consultas que se respondieron de manera exitosa, es decir, con la información solicitada por el usuario, y se comparan los promedios de ciclos de diálogo para cada caso de problemas.

## **CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS**

En el presente capítulo se presentan las conclusiones a las que se llegaron después de realizar este trabajo de tesis, culminando en la implementación del administrador de diálogo. Además, se incluyen las áreas de oportunidad a considerar para un desarrollo futuro.

### **5.1 Conclusiones**

Este trabajo tuvo como producto terminado un administrador de diálogo que se adaptó a la ILNBD del ITCM, y que colabora con la aclaración de consultas que presentan elipsis por insuficiencia de conocimiento, con lo cual se mejoró el desempeño de la interfaz a través de procesos de diálogo. Con lo anterior, se concluye que se cumplió el objetivo general de este proyecto.

Se puede asegurar el cumplimiento del objetivo a partir de los resultados obtenidos en la experimentación, donde el administrador de diálogo obtuvo un 100% en la prueba de desempeño, ya que para cada consulta del corpus proporcionó los datos esperados para la consulta en lenguaje natural (Tabla 4.6). Conviene destacar que la ILNBD sin administrador de diálogo logró responder correctamente el 10% de las consultas ingresadas que poseen problemas de elipsis semántica.

Además, en la prueba de comparación (Tabla 4.7), el administrador de diálogo de la ILNBD del ITCM obtuvo el 100% de respuestas correctas en contraste al 29% de la ILNBD del CENIDET, con lo que se concluye que el administrador de diálogo del presente trabajo ofrece resultados óptimos, al integrarse con la ILNBD del ITCM. Por otra parte, en cuanto a la comparación de

los ciclos de diálogo (Tabla 4.9), el número de ciclos de la ILNBD del ITCM es menor o igual al de la ILNBD del CENIDET para los casos de la prueba.

Por último, una contribución importante de esta tesis es una tipificación de problemas dobles que permitió tratar los problemas en un orden específico, logrando hacer más eficientes los procesos de diálogo para algunas consultas. En algunas consultas, reduce un caso de dos problemas a un caso de un problema, mientras que en otras, prescinde del diálogo de aclaración con el usuario, resolviendo la situación de elipsis de manera interna.

En resumen, la implementación del administrador de diálogo, con procesos de diálogo basados en la tipificación de problemas en consulta, mejoró el desempeño de la ILNBD del ITCM, permitiendo proporcionar al usuario los resultados esperados.

## **5.2 Trabajos futuros**

Las áreas de oportunidad que podrían ser revisadas en trabajos futuros a partir de este proyecto son:

- § Mejorar el diseño de las interfaces gráficas de usuario, de manera que faciliten aún más al usuario el diálogo con la ILNBD.
  
- § Modificar la identificación de columnas con los valores de búsqueda que ingrese el usuario en la consulta; es decir, localizar las columnas que coincidan con el tipo de dato y la extensión, por citar algunas características. Posiblemente, con el uso de expresiones regulares.

- § Incluir una opción que permita ingresar información adicional sobre las tablas y columnas de la BD que se esté consultando, es decir, del dominio específico de consulta, para ser utilizada por el administrador de diálogo. Estos datos podrían ser capturados por el administrador de la base de datos, para facilitar aún más la aclaración de consultas. Por ejemplo: Supóngase que se está consultando una BD con una tabla de *alumnos* que contiene las columnas *nombre*, *apellido paterno* y *apellido materno*, entre otras; se podría especificar en la información adicional que el término *nombre del alumno*, se refiere a la concatenación de las tres columnas antes mencionadas.
- § Diseñar procesos de diálogo para las funciones que no están implementadas en el administrador de diálogo de este trabajo, como funciones de agregación, operaciones de agrupación y subconsultas.

## ANEXO A. CORPUS DE CONSULTAS

En las tablas siguientes se presenta el corpus de consultas utilizadas en la experimentación, divididas por tipo de problema e identificadas según la base de datos (A – ATIS, G – Geobase, N – Northwind).

Tabla A.1 Consultas para pruebas con el problema 2.1

#	Consulta	BD
1	¿Cuáles son las tarifas? <i>SELECT fare.fare_code, fare.one_way_cost, fare.rnd_trip_cost FROM fare</i>	A
2	¿Cuáles son los fletadores? <i>SELECT Shippers.ShipperID, Shippers.CompanyName FROM Shippers</i>	N
3	¿Cuáles son los transportes? <i>SELECT transport.transport_code, transport.transport_desc FROM transport</i>	A
4	Lista los estados. <i>SELECT state.state_code, state.state_name FROM state</i>	A
5	Lista los aviones. <i>SELECT aircraft.aircraft_type, aircraft.category FROM aircraft</i>	A
6	Lista los territorios. <i>SELECT Territories.TerritoryID, Territories.TerritoryDescription FROM Territories</i>	N
7	Muestra los alimentos. <i>SELECT food_service.meal_class, food_service.meal_description FROM food_service</i>	A
8	Muestra todas las órdenes. <i>SELECT Orders.OrderID, Orders.OrderDate FROM Orders</i>	N
9	Muestra todos los vuelos. <i>SELECT flight.flight_number, flight.from_airport, flight.to_airport FROM flight</i>	A
10	Despliega todos los empleados. <i>SELECT Employees.EmployeeID, Employees.LastName, Employees.FirstName FROM Employees</i>	N

Tabla A.2 Consultas para pruebas con el problema 3.1

#	Consulta	BD
1	¿Cuál es el peso de un D10? <i>SELECT aircraft.weight FROM aircraft WHERE aircraft.aircraft_code LIKE 'D10'</i>	A
2	¿Cuál es el punto más alto en Colorado? <i>SELECT HighLow.highest_point FROM HighLow WHERE HighLow.state_name LIKE 'Colorado'</i>	G
3	¿Cuál es la capital de California? <i>SELECT State.capital FROM State WHERE State.state_name LIKE 'California'</i>	G
4	¿Cuál es la extensión de alas en un 72S? <i>SELECT aircraft.wing_span FROM aircraft WHERE aircraft.aircraft_code LIKE '72S'</i>	A
5	¿Cuál es la longitud del Mississippi? <i>SELECT River.length FROM River WHERE River.river_name LIKE 'Mississippi'</i>	G
6	¿Cuál es la velocidad de un JET? <i>SELECT aircraft.cruising_speed FROM aircraft WHERE aircraft.category LIKE 'JET'</i>	A
7	Dame los proveedores de Frankfurt. <i>SELECT Suppliers.CompanyName FROM Suppliers WHERE Suppliers.City LIKE 'Frankfurt'</i>	N
8	Lista el número de asientos en el M80. <i>SELECT aircraft.capacity FROM aircraft WHERE aircraft.aircraft_code LIKE 'M80'</i>	A
9	¿Qué ciudades están localizadas en Pennsylvania? <i>SELECT City.city_name FROM City WHERE City.state_name LIKE 'Pennsylvania'</i>	G
10	¿Qué montañas están en Alaska? <i>SELECT Mountain.mountain_name FROM Mountain WHERE Mountain.state_name LIKE 'Alaska'</i>	G

Tabla A.3 Consultas para pruebas con el problema 4.1

#	Consulta	BD
1	¿Cuáles son las poblaciones? <i>SELECT State.population FROM State</i>	G
2	¿Cuáles son los códigos de aeropuerto? <i>SELECT airport.airport_code FROM airport</i>	A
3	Despliega las abreviaturas de estado. <i>SELECT City.state_abbreviation FROM City</i>	G

Tabla A.3 Consultas para pruebas con el problema 4.1 (continuación)

4	Dame los tiempos de viaje. <i>SELECT flight.time_elapsed FROM flight</i>	A
5	Lista las áreas. <i>SELECT State.area FROM State</i>	G
6	Lista las escalas. <i>SELECT fconnection.stops FROM fconnection</i>	A
7	Lista los países. <i>SELECT Customers.Country FROM Customers</i>	N
8	Lista todos los productos y precios. <i>SELECT Products.ProductName, Products.UnitPrice FROM Products</i>	N
9	Muestra los teléfonos. <i>SELECT Suppliers.Phone FROM Suppliers</i>	N
10	Muéstrame las horas de salida. <i>SELECT flight.departure_time FROM flight</i>	A

Tabla A.4 Consultas para pruebas con el problema 5.1

#	Consulta	BD
1	¿Cuáles lagos tienen un área mayor a 10000? <i>SELECT Lake.lake_name FROM Lake WHERE Lake.area&gt;10000</i>	G
2	¿Cuáles son las ciudades con población menor a 10000? <i>SELECT City.city_name FROM City WHERE City.population&lt;10000</i>	G
3	¿Cuál es la restricción para el código de tarifa 7100515? <i>SELECT restriction.restrict_code FROM restriction, fare WHERE restriction.restrict_code = fare.restrict_code AND fare.fare_code = 7100515</i>	A
4	Dame la ubicación del código de aeropuerto BOS. <i>SELECT airport.location FROM airport WHERE airport.airport_code LIKE 'BOS'</i>	A
5	Lista la categoría de aeroplano para el código WLH. <i>SELECT aircraft.category FROM aircraft WHERE aircraft.aircraft_code LIKE 'WLH'</i>	A
6	Lista las ciudades con abreviatura de estado CA <i>SELECT City.city_name FROM City WHERE City.state_abbreviation LIKE 'CA'</i>	G
7	Lista los clientes de la región WA. <i>SELECT Customers.CompanyName FROM Customers WHERE Customers.Region LIKE 'WA'</i>	N

Tabla A.4 Consultas para pruebas con el problema 5.1 (continuación)

8	Muestra la clase de código para el código de vuelo 102135. <i>SELECT flight.class_string FROM flight WHERE flight.flight_code = 102135</i>	A
9	Muestra los productos con precio menor a 10. <i>SELECT Products.ProductName FROM Products WHERE Products.UnitPrice&lt;10</i>	N
10	¿Quién es el proveedor de la ciudad de Osaka? <i>SELECT Suppliers.CompanyName FROM Suppliers WHERE Suppliers.City LIKE 'Osaka'</i>	N

Tabla A.5 Consultas para pruebas con los problemas 2.1 y 3.1

#	Consulta	BD
1	¿Cuál es el empleado que es Sales Representative? <i>SELECT Employees.FirstName, Employees.LastName FROM Employees WHERE Employees.Title LIKE 'Sales Representative'</i>	N
2	¿Cuáles son las órdenes de VINET? <i>SELECT Orders.OrderID, Orders.OrderDate FROM Orders WHERE Orders.CustomerID LIKE 'VINET'</i>	N
3	Lista los aviones TURBOPROP. <i>SELECT aircraft.aircraft_code, aircraft.aircraft_type FROM aircraft WHERE aircraft.category LIKE 'TURBOPROP'</i>	A
4	Lista las órdenes de USA. <i>SELECT Orders.OrderID, Orders.ShipCity FROM Orders WHERE Orders.ShipCountry LIKE 'USA'</i>	N
5	Lista las tarifas QW. <i>SELECT fare.fare_code, fare.one_way_cost, fare.rnd_trip_cost FROM fare WHERE fare.fare_class LIKE 'QW'</i>	A
6	Lista los aviones con tarifas AP/80. <i>SELECT aircraft.aircraft_code, aircraft.aircraft_type FROM aircraft, fare, flight, flight_fare WHERE aircraft.aircraft_code = flight.aircraft_code AND flight.flight_code = flight_fare.flight_code AND flight_fare.fare_code = fare.fare_code AND fare.restrict_code LIKE 'AP/80'</i>	A
7	Muestra los empleados en el territorio de Phoenix. <i>SELECT Employees.EmployeeID, Employees.LastName FROM Employees, Territories, EmployeeTerritories WHERE Employees.EmployeeID = EmployeeTerritories.EmployeeID AND EmployeeTerritories.TerritoryID = Territories.TerritoryID AND Territories.TerritoryDescription LIKE 'Phoenix'</i>	N

Tabla A.5 Consultas para pruebas con los problemas 2.1 y 3.1 (continuación)

8	Muestra los territorios del 4. <i>SELECT Territories.TerritoryDescription FROM Territories WHERE Territories.RegionID = 4</i>	N
9	Muéstrame los alimentos B/S. <i>SELECT food_service.meal_class, food_service.meal_description FROM food_service WHERE food_service.meal_code LIKE 'B/S'</i>	A
10	Muéstrame las tarifas para el vuelo 106245. <i>SELECT fare.one_way_cost, fare.rnd_trip_cost FROM fare, flight, flight_fare WHERE fare.fare_code = flight_fare.fare_code AND flight_fare.flight_code = flight.flight_code AND flight.flight_code = 106245</i>	A

Tabla A.6 Consultas para pruebas con los problemas 2.1 y 4.1

#	Consulta	BD
1	¿Cuáles son las horas de salida y los alimentos? <i>SELECT flight.departure_time, food_service.meal_class, food_service.meal_description FROM food_service, flight WHERE food_service.meal_code = flight.meal_code</i>	A
2	¿Cuáles son las tarifas y las horas de llegada? <i>SELECT flight.arrival_time, fare.one_way_cost, fare.rnd_trip_cost FROM fare, flight, flight_fare WHERE fare.fare_code = flight_fare.fare_code AND flight_fare.flight_code = flight.flight_code</i>	A
3	Dame las tarifas y los tiempos de viaje. <i>SELECT flight.time_elapsed, fare.one_way_cost, fare.rnd_trip_cost FROM fare, flight, flight_fare WHERE fare.fare_code = flight_fare.fare_code AND flight_fare.flight_code = flight.flight_code</i>	A
4	Dame los territorios y los países. <i>SELECT Employees.Country, Territories.TerritoryID, Territories.TerritoryDescription FROM Territories, Employees, EmployeeTerritories WHERE Territories.TerritoryID = EmployeeTerritories.TerritoryID AND EmployeeTerritories.EmployeeID = Employees.EmployeeID</i>	N
5	Lista precios y categorías. <i>SELECT Products.UnitPrice, Categories.Description FROM Categories, Products WHERE Categories.CategoryID = Products.CategoryID</i>	N
6	Lista las órdenes y los precios. <i>SELECT OrderDetails.UnitPrice, Orders.OrderID, Orders.OrderDate FROM Orders, OrderDetails WHERE Orders.OrderID = OrderDetails.OrderID</i>	N

Tabla A.6 Consultas para pruebas con los problemas 2.1 y 4.1 (continuación)

7	Lista vuelos y códigos. <i>SELECT aircraft.aircraft_code, flight.flight_number, flight.from_airport, flight.to_airport FROM flight, aircraft WHERE flight.aircraft_code = aircraft.aircraft_code</i>	A
8	Muestra todas las tarifas y las escalas. <i>SELECT restriction.stopovers, fare.one_way_cost, fare.rnd_trip_cost FROM fare, restriction WHERE fare.restrict_code = restriction.restrict_code</i>	A
9	Muestra transportes y códigos de aeropuerto. <i>SELECT ground_service.airport_code, transport.transport_code, transport.transport_desc FROM transport, ground_service WHERE transport.transport_code = ground_service.transport_code</i>	A
10	Muéstrame aviones y tiempos de viaje. <i>SELECT flight.time_elapsed, aircraft.category, aircraft.capacity FROM aircraft, flight WHERE aircraft.aircraft_code = flight.aircraft_code</i>	A

Tabla A.7 Consultas para pruebas con los problemas 2.1 y 5.1

#	Consulta	BD
1	¿Cuáles aviones tienen tiempos de viaje menores a 25? <i>SELECT aircraft.aircraft_code, aircraft.aircraft_type FROM aircraft, flight WHERE aircraft.aircraft_code = flight.aircraft_code AND flight.time_elapsed &lt; 25</i>	A
2	¿Cuáles órdenes tienen descuentos del 0.5? <i>SELECT Orders.OrderID, Orders.OrderDate FROM Orders, OrderDetails WHERE Orders.OrderID = OrderDetails.OrderID AND OrderDetails.Discount = 0.5</i>	N
3	¿Cuáles son las tarifas en hora de salida a las 700? <i>SELECT fare.fare_class, fare.one_way_cost FROM fare, flight, flight_fare WHERE fare.fare_code = flight_fare.fare_code AND flight_fare.flight_code = flight.flight_code AND flight.departure_time = 700</i>	A
4	¿Cuáles son los empleados en la ciudad de London? <i>SELECT Employees.EmployeeID, Employees.FirstName, Employees.LastName, Employees.Title FROM Employees WHERE Employees.City LIKE 'London'</i>	N
5	¿Cuáles son los transportes para el código de aeropuerto SFO? <i>SELECT transport.transport_code, transport.transport_desc FROM transport, ground_service WHERE transport.transport_code = ground_service.transport_code AND ground_service.airport_code LIKE 'SFO'</i>	A

Tabla A.7 Consultas para pruebas con los problemas 2.1 y 5.1 (continuación)

6	¿Cuáles son los vuelos con código L10? <i>SELECT flight.flight_number, flight.from_airport, flight.to_airport FROM flight, aircraft WHERE flight.aircraft_code = aircraft.aircraft_code AND aircraft.aircraft_code LIKE 'L10'</i>	A
7	Lista las tarifas para tiempos de viaje mayor a 500. <i>SELECT fare.one_way_cost FROM fare, flight, flight_fare WHERE fare.fare_code = flight_fare.fare_code AND flight_fare.flight_code = flight.flight_code AND flight.time_elapsed&gt;500</i>	A
8	Lista los alimentos para la hora de llegada a las 1000. <i>SELECT food_service.meal_class, food_service.meal_description FROM food_service, flight WHERE food_service.meal_code = flight.meal_code AND flight.arrival_time = 1000</i>	A
9	Lista los territorios de la región WA. <i>SELECT Territories.TerritoryID, Territories.TerritoryDescription FROM Territories, Employees, EmployeeTerritories WHERE Territories.TerritoryID = EmployeeTerritories.TerritoryID AND EmployeeTerritories.EmployeeID = Employees.EmployeeID AND Employees.Region LIKE 'WA'</i>	N
10	Muestra los vuelos con número de escalas mayor a 2. <i>SELECT flight.flight_number, flight.airline_code FROM flight WHERE flight.stops&gt;2</i>	A

Tabla A.8 Consultas para pruebas con los problemas 4.1 y 3.1

#	Consulta	BD
1	¿Cuál es el área de Alaska? <i>SELECT State.area FROM State WHERE State.state_name LIKE 'Alaska'</i>	G
2	¿Cuáles son los códigos de tarifa para el vuelo 351? <i>SELECT flight_fare.fare_code FROM flight_fare, flight WHERE flight_fare.flight_code = flight.flight_code AND flight.flight_number = 351</i>	A
3	¿Cuál es el código de vuelo del 140? <i>SELECT flight.flight_code FROM flight WHERE flight.flight_number = 140</i>	A
4	¿Cuál es el código de ciudad para Baltimore? <i>SELECT city.city_code FROM city WHERE city.city_name LIKE 'Baltimore'</i>	A
5	¿Cuál es el precio de un Chai? <i>SELECT Products.UnitPrice FROM Products WHERE Products.ProductName LIKE 'Chai'</i>	N

Tabla A.8 Consultas para pruebas con los problemas 4.1 y 3.1 (continuación)

6	¿Cuál es la población de Boston? <i>SELECT City.population FROM City WHERE City.city_name LIKE 'Boston'</i>	G
7	Dame el tiempo de viaje del 644. <i>SELECT flight.time_elapsed FROM flight WHERE flight.flight_number = 644</i>	A
8	¿Cuál es el número de escalas en el vuelo 21? <i>SELECT flight.stops FROM flight WHERE flight.flight_number = 21</i>	A
9	Lista los precios de la 4. <i>SELECT Products.UnitPrice FROM Products WHERE Products.CategoryID = 4</i>	N
10	Muestra el teléfono de United Package. <i>SELECT Shippers.Phone FROM Shippers WHERE Shippers.CompanyName LIKE 'United Package'</i>	N

Tabla A.9 Consultas para pruebas con los problemas 4.1 y 5.1

#	Consulta	BD
1	¿Cuál es el aeropuerto destino para la hora de salida de las 900? <i>SELECT flight.to_airport FROM flight WHERE flight.departure_time = 900</i>	A
2	¿Cuál es el área del que tiene abreviatura TX? <i>SELECT State.area FROM State WHERE State.abbreviation LIKE 'TX'</i>	G
3	¿Cuál es el aeropuerto destino para el de código J31? <i>SELECT flight.to_airport FROM flight, aircraft WHERE flight.aircraft_code = aircraft.aircraft_code AND aircraft.aircraft_code LIKE 'J31'</i>	A
4	¿Cuál es el tiempo de viaje con escalas mayor a 2? <i>SELECT fconnection.time_elapsed FROM fconnection WHERE fconnection.stops&gt;2</i>	A
5	¿Cuáles son las ciudades de la región WA? <i>SELECT Employees.City FROM Employees WHERE Employees.Region LIKE 'WA'</i>	N
6	¿Cuáles son los tiempos de viaje para las horas de salida después de las 2000? <i>SELECT flight.time_elapsed FROM flight WHERE flight.departure_time&gt;2000</i>	A
7	Lista las poblaciones de las áreas menores a 5000. <i>SELECT State.population FROM State WHERE State.area&lt;5000</i>	G

Tabla A.9 Consultas para pruebas con los problemas 4.1 y 5.1 (continuación)

8	Lista las cantidades para los que tienen precio mayor a 100. <i>SELECT Products.QuantityPerUnit FROM Products WHERE Products.UnitPrice&gt;100</i>	N
9	Lista los códigos de aeropuerto para el código de estado CA. <i>SELECT airport.airport_code FROM airport WHERE airport.state_code LIKE 'CA'</i>	A
10	Muestra los teléfonos de los que están en el país UK. <i>SELECT Suppliers.Phone FROM Suppliers WHERE Suppliers.Country LIKE 'UK'</i>	N

Tabla A.10 Consultas para pruebas con los problemas 3.1 y 5.1

#	Consulta	BD
1	¿Cuál es la clase para una tarifa AP/57 con hora de salida antes de las 800? <i>SELECT compound_class.class_type FROM compound_class, fare, flight, flight_fare WHERE compound_class.fare_class = fare.fare_class AND fare.fare_code = flight_fare.fare_code AND flight_fare.flight_code = flight.flight_code AND flight.departure_time&lt;800 AND fare.restrict_code LIKE 'AP/57'</i>	A
2	¿Cuál es la tarifa redonda para los vuelos US con un tiempo de viaje menor a 50? <i>SELECT fare.rnd_trip_cost FROM fare, flight, flight_fare WHERE fare.fare_code = flight_fare.fare_code AND flight_fare.flight_code = flight.flight_code AND flight.time_elapsed&lt;50 AND flight.airline_code LIKE 'US'</i>	A
3	¿Cuáles son las restricciones para la tarifa 7100144 y el código de vuelo 106373? <i>SELECT restriction.restrict_code FROM restriction, fare, flight_fare WHERE restriction.restrict_code = fare.restrict_code AND fare.fare_code = flight_fare.fare_code AND flight_fare.flight_code = 106373 AND fare.fare_code = 7100144</i>	A
4	¿Cuáles son los identificadores de órdenes procesadas por empleados en USA que viven en la ciudad de Seattle? <i>SELECT Orders.OrderID FROM Orders, Employees WHERE Orders.EmployeeID = Employees.EmployeeID AND Employees.City LIKE 'Seattle' AND Employees.Country LIKE 'USA'</i>	N

Tabla A.10 Consultas para pruebas con los problemas 3.1 y 5.1 (continuación)

5	<p>Lista las ciudades con código de aeropuerto DFW y que tengan transporte R.</p> <pre>SELECT city.city_name FROM city, airport_service, transport, ground_service WHERE city.city_code = airport_service.city_code AND airport_service.city_code = ground_service.city_code AND airport_service.airport_code = ground_service.airport_code AND ground_service.transport_code = transport.transport_code AND transport.transport_code LIKE 'R' AND airport_service.airport_code LIKE 'DFW'</pre>	A
6	<p>Lista los aeropuertos para el vuelo 105805 con código de aerolínea DL.</p> <pre>SELECT airport.airport_name FROM airport, flight WHERE airport.airport_code = flight.from_airport AND airport.airport_code = flight.to_airport AND flight.airline_code LIKE 'DL' AND flight.flight_code = 105805</pre>	A
7	<p>Lista los clientes con órdenes enviadas a Madrid con precios mayores a 100.</p> <pre>SELECT Customers.CompanyName FROM Customers, Orders, OrderDetails WHERE Customers.CustomerID = Orders.CustomerID AND Orders.OrderID = OrderDetails.OrderID AND OrderDetails.UnitPrice &gt; 100 AND Orders.ShipCity LIKE 'Madrid'</pre>	N
8	<p>Lista los números de vuelo con alimentos FIRST y número de escalas mayor a 1.</p> <pre>SELECT flight.flight_number FROM flight, food_service WHERE flight.meal_code = food_service.meal_code AND flight.stops &gt; 1 AND food_service.meal_class LIKE 'FIRST'</pre>	A
9	<p>Lista los productos en las órdenes del 9 con precios menores a 10.</p> <pre>SELECT Products.ProductName FROM Products, Orders, OrderDetails WHERE Products.ProductID = OrderDetails.ProductID AND OrderDetails.OrderID = Orders.OrderID AND Orders.OrderID = OrderDetails.OrderID AND OrderDetails.UnitPrice &lt; 10 AND Orders.EmployeeID = 9</pre>	N
10	<p>Dame los vuelos con código de aerolínea LH en aviones JET.</p> <pre>SELECT flight.flight_number, flight.from_airport, flight.to_airport FROM flight, aircraft WHERE flight.aircraft_code = aircraft.aircraft_code AND flight.airline_code LIKE 'LH' AND aircraft.category LIKE 'JET'</pre>	A

## ANEXO B. DESCRIPCIÓN DE LA BASE DE DATOS ATIS

ATIS (Air Travel Information Service) es una base de datos relacional que contiene registros de información sobre vuelos. Esta base de datos contiene 25 tablas y 118 columnas. Este anexo presenta el esquema y los detalles de las tablas de la base de datos.

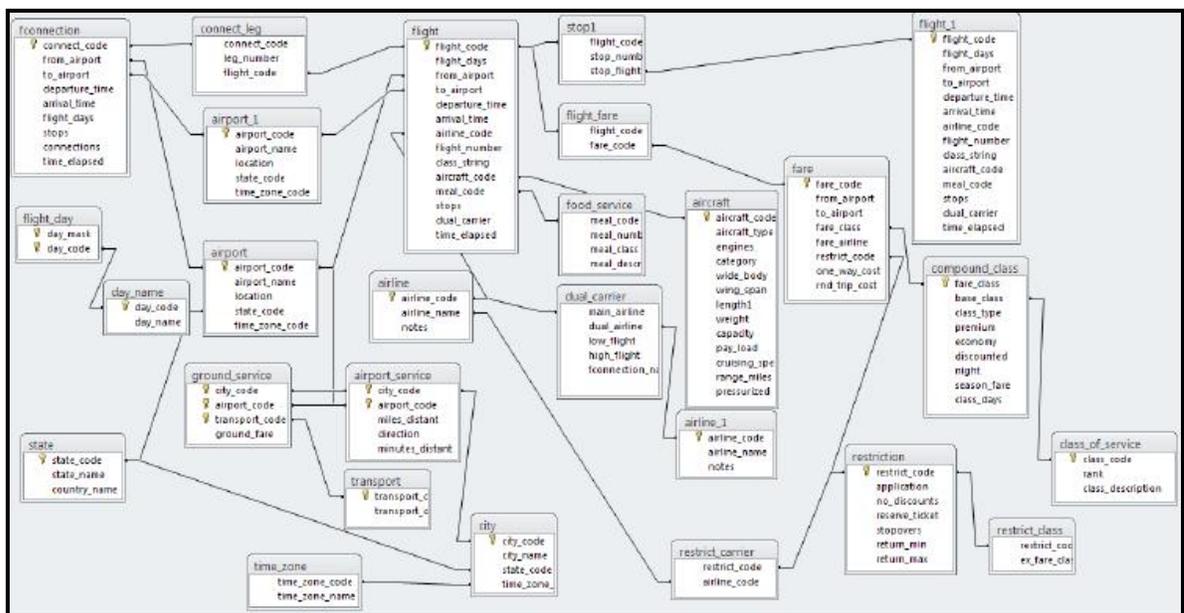


Figura B.1 Esquema de la base de datos ATIS

Tabla: *aircraft* (Descripción: avión)

Columna	Tipo de dato	Descripción
aircraft_code	texto	código
aircraft_type	texto	tipo de aeronave
capacity	entero	número de asientos
category	texto	categoría de aeroplano
cruising_speed	entero	velocidad
engines	entero	número de motores
length1	flotante	tamaño de equipo

pay_load	entero	carga
pressurized	texto	presurización
range_miles	entero	longitud de vuelo
weight	entero	peso
wide_body	texto	ancho de cuerpo
wing_span	flotante	extensión de alas

**Llave primaria:** aircraft\_code

**Tabla: *airline* (Descripción: aerolínea)**

Columna	Tipo de dato	Descripción
airline_code	texto	código de aerolínea
airline_name	texto	aerolínea
notes	texto	notas

**Llave primaria:** airline\_code

**Tabla *airport* (Descripción: aeropuerto)**

Columna	Tipo de dato	Descripción
airport_code	texto	código de aeropuerto
airport_name	texto	aeropuerto
location	texto	ubicación
state_code	texto	código de estado
time_zone_code	texto	código de zona de horario

**Llave primaria:** airport\_code

**Llave foránea:** airport (state\_code) – state (state\_code)

**Tabla *airport\_service* (Descripción: servicio de aeropuerto)**

Columna	Tipo de dato	Descripción
airport_code	texto	código de aeropuerto
city_code	texto	código de ciudad
direction	texto	dirección
miles_distant	flotante	distancia en millas
minutes_distant	entero	distancia en minutos

---

<b>Llave primaria:</b>	city_code, airport_code
<b>Llave foránea:</b>	airport_service (airport_code) – airport (airport_code) airport_service (airport_code) – ground_service (airport_code) airport_service (city_code) – ground_service (airport_service) airport_service (city_code) – city (city_code)

---

**Tabla *city* (Descripción: ciudad)**

Columna	Tipo de dato	Descripción
city_code	texto	código de ciudad
city_name	texto	ciudad
state_code	texto	código de estado
time_zone_code	texto	código de zona de horario
<b>Llave primaria:</b>	city_code	
<b>Llave foránea:</b>	city (state_code) – state (state_code) city (time_zone_code) – time_zone (time_zone_code)	

---

**Tabla *class\_of\_service* (Descripción: clases de servicio)**

Columna	Tipo de dato	Descripción
class_code	texto	código de clase de servicio
class_description	texto	clase de servicio
rank	entero	rango
<b>Llave primaria:</b>	class_code	

---

**Tabla *code\_description* (Descripción: códigos)**

Columna	Tipo de dato	Descripción
code	texto	código
description	texto	descripción de código
<b>Llave primaria:</b>	code	

---

**Tabla *compound\_class* (Descripción: clases de tarifa)**

Columna	Tipo de dato	Descripción
fare_class	texto	código de clase de tarifa
base_class	texto	clase de clase de servicio

class_days	texto	días
class_type	texto	clase
discounted	texto	descuento
economy	texto	económica
night	texto	de noche
premium	texto	primera
season_fare	texto	de temporada

**Llave primaria:** fare\_class

**Llave foránea:** compound\_class (base\_class) – class\_of\_service (class\_code)

**Tabla *connect\_leg* (Descripción: segmentos de conexiones)**

Columna	Tipo de dato	Descripción
connect_code	entero	código de conexión
flight_code	entero	código de vuelo
leg_number	entero	número de segmento
<b>Llave primaria:</b> connect_code, leg_number		
<b>Llave foránea:</b> connect_leg (connect_code) – fconnection (connect_code) connect_leg (flight_code) – flight (flight_code)		

**Tabla *day\_name* (Descripción: días)**

Columna	Tipo de dato	Descripción
day_code	entero	código de día
day_name	texto	nombre de día
<b>Llave primaria:</b> day_code		

**Tabla *dual\_carrier* (Descripción: empresas duales)**

Columna	Tipo de dato	Descripción
dual_airline	texto	código de empresa dual
fconnection_name	texto	nombre de conexión
high_flight	entero	número mayor de vuelo
low_flight	entero	número menor de vuelo
main_airline	texto	código de empresa principal

---

<b>Llave primaria:</b>	dual_airline, main_airline
<b>Llave foránea:</b>	dual_carrier (dual_airline) – airline (airline_code) dual_carrier (main_airline) – airline (airline_code)

---

**Tabla *fare* (Descripción: tarifa)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
fare_code	entero	código de tarifa
fare_airline	texto	tarifa de aerolínea
fare_class	texto	clase de tarifa
from_airport	texto	aeropuerto de origen
one_way_cost	flotante	tarifa sencilla
restrict_code	texto	código de restricción
rnd_trip_cost	flotante	tarifa redonda
to_airport	texto	aeropuerto destino
<b>Llave primaria:</b>	fare_code	
<b>Llave foránea:</b>	fare (restrict_code) – restriction (restrict_code) fare (fare_class) – compound_class (fare_class)	

---

**Tabla *fconnection* (descripción: conexión)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
connect_code	entero	código de conexión
arrival_time	entero	horas de llegada
connections	entero	conexiones
departure_time	entero	horas de salida
flight_days	texto	días de vuelo
from_airport	texto	aeropuerto de origen
stops	entero	escalas
time_elapsed	entero	tiempo de viaje
<b>Llave primaria:</b>	connect_code	
<b>Llave foránea:</b>	fconnection (to_airport) – airport (airport_code) fconnection (from_airport) – airport (airport_code)	

---

**Tabla: *flight* (Descripción: vuelo)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
flight_code	entero	código de vuelo
aircraft_code	texto	código de avión
airline_code	texto	código de aerolínea
arrival_time	entero	horas de llegada
class_string	texto	clase de código
departure_time	entero	horas de salida
dual_carrier	texto	empresa dual
flight_days	texto	días
flight_number	entero	número de vuelo
from_airport	texto	aeropuerto de origen
meal_code	texto	código de comida
stops	entero	número de escalas
time_elapsed	entero	tiempo de viaje
to_airport	texto	aeropuerto destino
<b>Llave primaria:</b>	flight_code	
<b>Llave foránea:</b>	flight (aircraft_code) – aircraft (aircraft_code) flight (airline_code) – airline (airline_code) flight (from_airport) – airport (airport_code) flight (to_airport) – airport (airport_code)	

**Tabla *flight\_day* (Descripción: días de vuelo)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
day_code	entero	código de día
day_mask	texto	máscara de días
<b>Llave primaria:</b>	day_mask, day_code	

**Tabla *flight\_fare* (Descripción: tarifas de vuelos)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
fare_code	entero	código de tarifa
flight_code	entero	código de vuelo
<b>Llave primaria:</b>	fare_code, flight_code	

**Llave foránea:** flight\_fare (fare\_code) – fare (fare\_code)  
flight\_fare (flight\_code) – flight (flight\_code)

---

**Tabla *food\_service* (Descripción: alimentos)**

Columna	Tipo de dato	Descripción
meal_code	texto	código de comida
meal_class	texto	clase de comida
meal_number	entero	número de comida
meal_description	texto	descripción de comida
<b>Llave primaria:</b> meal_code, meal_class, meal_number		
<b>Llave foránea:</b> food_service (meal_code) – flight (meal_code)		

**Tabla *ground\_service* (Descripción: servicios de transporte terrestre)**

Columna	Tipo de dato	Descripción
transport_code	texto	código de transporte
city_code	texto	código de ciudad
airport_code	texto	código de aeropuerto
ground_fare	flotante	tarifa terrestre
<b>Llave primaria:</b> city_code, airport_code, transport_code		
<b>Llave foránea:</b> ground_service (transport_code) – transport (transport_code)		

**Tabla *restrict\_carrier* (Descripción: restricciones de aerolínea)**

Columna	Tipo de dato	Descripción
restrict_code	texto	código de restricción
airline_code	texto	código de aerolínea
<b>Llave primaria:</b> restrict_code, airline_code		
<b>Llave foránea:</b> restrict_carrier (restrict_code) – restriction (restrict_code)		

**Tabla *restrict\_class* (Descripción: restricciones de clases)**

Columna	Tipo de dato	Descripción
restrict_code	texto	código de restricción
ex_fare_class	texto	tarifa de clase

---

**Llave primaria:** restrict\_code, ex\_fare\_class

**Llave foránea:** restrict\_class (restrict\_code) – restriction (restrict\_code)

---

**Tabla *restriction* (Descripción: restricción)**

Columna	Tipo de dato	Descripción
restrict_code	texto	restricción
application	texto	aplicación
no_discounts	texto	descuentos no aplicables
reserve_ticket	entero	boleto en reserva
return_max	entero	máximo de permanencia
return_min	entero	mínimo de permanencia
stopovers	texto	Escalas

**Llave primaria:** restrict\_code

---

**Tabla *state* (Descripción: estados)**

Columna	Tipo de dato	Descripción
state_code	texto	código de estado
state_name	texto	nombre de estado
country_name	texto	nombre de país

**Llave primaria:** state\_code

---

**Tabla *stop1* (Descripción: escalas)**

Columna	Tipo de dato	Descripción
flight_code	entero	código de vuelo
stop_flight	entero	vuelo de escala
stop_number	entero	número de escala

**Llave primaria:** flight\_code, stop\_number

**Llave foránea:** stop1 (flight\_code) – flight (flight\_code)  
stop1 (stop\_flight) – flight (flight\_code)

---

**Tabla *time\_zone* (Descripción: zonas de horarios)**

Columna	Tipo de dato	Descripción
---------	--------------	-------------

---

time_zone_code	texto	código de zona de horario
time_zone_name	texto	nombre de zona de horario
<b>Llave primaria:</b>	time_zone_code	
<b>Llave foránea:</b>	time_zone (time_zone_code) – city (time_zone_code)	

**Tabla *transport* (Descripción: transporte)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
transport_code	texto	código de transporte
transport_desc	texto	descripción de transporte
<b>Llave primaria:</b>	transport_code	

## ANEXO C. DESCRIPCIÓN DE LA BASE DE DATOS NORTHWIND

Northwind es una base de datos relacional de prueba con información de un negocio de comercio. Esta base de datos contiene 13 tablas y 88 columnas. En este anexo se presenta el esquema y los detalles de las tablas de la base de datos.

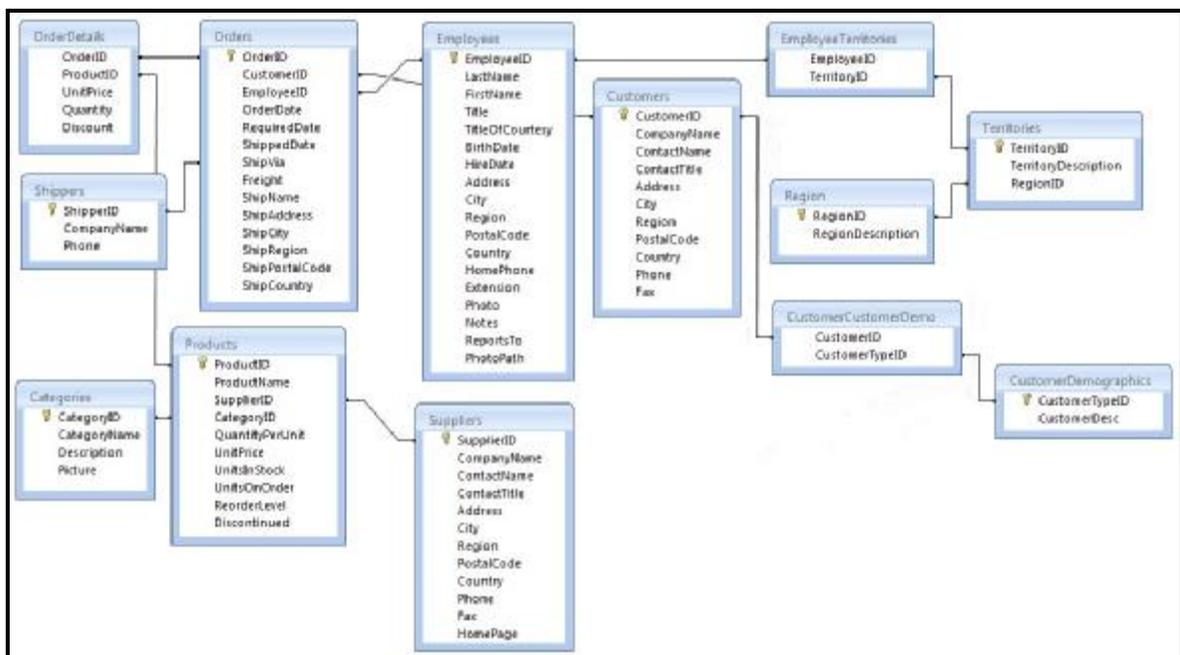


Figura C.1 Esquema de la base de datos Northwind

Tabla: **Categories** (Descripción: categorías)

Columna	Tipo de dato	Descripción
CategoryID	entero	identificador de categoría
CategoryName	texto	nombre de categoría
Description	texto	descripción
Picture	binario	imagen
<b>Llave primaria:</b>	CategoryID	

**Tabla: *Customers* (Descripción: clientes)**

Columna	Tipo de dato	Descripción
CustomerID	texto	identificador de cliente
Address	texto	dirección
City	texto	ciudad
CompanyName	texto	clientes
ContactName	texto	nombre de contacto
ContactTitle	texto	puesto de contacto
Country	texto	país
Fax	texto	fax
Phone	texto	teléfono
PostalCode	texto	código postal
Region	texto	región
<b>Llave primaria:</b>	CustomerID	

**Tabla: *CustomerCustomerDemo* (Descripción: demo)**

Columna	Tipo de dato	Descripción
CustomerID	texto	identificador de cliente
CustomerTypeID	texto	identificador de tipo de cliente
<b>Llave primaria:</b>	CustomerID, CustomerTypeID	
<b>Llave foránea:</b>	CustomerCustomerDemo (CustomerID) – Customers (CustomerID) CustomerCustomerDemo (CustomerTypeID) – CustomersDemographics (CustomerTypeID)	

**Tabla: *CustomerDemographics* (Descripción: demografías)**

Columna	Tipo de dato	Descripción
CustomerTypeID	texto	identificador de tipo de cliente
CustomerDesc	texto	descripción de cliente
<b>Llave primaria:</b>	CustomerTypeID	

**Tabla: *Employees* (Descripción: empleados)**

Columna	Tipo de dato	Descripción
EmployeeID	entero	identificador de empleado

Address	texto	dirección
BirthDate	fecha	fecha de nacimiento
City	texto	ciudad
Country	texto	país
Extension	texto	extensión
FirstName	texto	nombre de empleado
HireDate	fecha	fecha de contratación
HomePhone	texto	teléfono
LastName	texto	apellido de empleado
Notes	texto	notas
Photo	binario	fotografía
PhotoPath	texto	ruta de fotografía
PostalCode	texto	código postal
Region	texto	región
ReportsTo	entero	reportes para
Title	texto	puesto
TitleOfCourtesy	texto	título de cortesía
<hr/>		
<b>Llave primaria:</b>	EmployeeID	

**Tabla: *EmployeeTerritories* (Descripción: territorios de empleados)**

Columna	Tipo de dato	Descripción
EmployeeID	entero	identificador de empleado
TerritoryID	texto	identificador de territorio
<hr/>		
<b>Llave primaria:</b>	EmployeeID, TerritoryID	
<b>Llave foránea:</b>	EmployeeTerritories (EmployeeID) – Employees (EmployeeID) EmployeeTerritories (TerritoryID) – Territories (TerritoryID)	

**Tabla: *Orders* (Descripción: órdenes)**

Columna	Tipo de dato	Descripción
OrderID	entero	identificador de orden
CustomerID	texto	identificador de cliente
EmployeeID	entero	identificador de empleado
Freight	moneda	monto de flete

OrderDate	fecha	fecha de orden
RequiredDate	fecha	fecha de requisición
ShipAddress	texto	dirección de navío
ShipCity	texto	ciudad de navío
ShipCountry	texto	país de navío
ShipName	texto	nombre de navío
ShippedDate	fecha	fecha de embarcación
ShipPostalCode	texto	código postal de navío
ShipRegion	texto	región de navío
ShipVia	entero	método de embarcación
<b>Llave primaria:</b>	OrderID	
<b>Llave foránea:</b>	Orders (CustomerID) – Customers (CustomerID) Orders (EmployeeID) – Employees (EmployeeID) Orders (ShipVia) – Shippers (ShipperID)	

**Tabla: *OrderDetails* (Descripción: detalles de orden)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
Discount	flotante	descuento
OrderID	entero	identificador
ProductID	entero	identificador de producto
Quantity	entero	cantidad
UnitPrice	moneda	precio
<b>Llave primaria:</b>	OrderID, ProductID	
<b>Llave foránea:</b>	OrderDetails (OrderID) – Orders (OrderID) OrderDetails (ProductID) – Products (ProductID)	

**Tabla: *Products* (Descripción: productos)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
ProductID	entero	identificador de producto
CategoryID	entero	identificador de categoría
Discontinued	booleano	descuento
ProductName	texto	producto
QuantityPerUnit	texto	cantidad

ReorderLevel	entero	nivel de nuevo pedido
SupplierID	entero	identificador de proveedor
UnitPrice	moneda	precio
UnitsInStock	entero	unidades en reserva
UnitsOnOrder	entero	unidades en pedido
<b>Llave primaria:</b>	ProductID	
<b>Llave foránea:</b>	Products (CategoryID) – Categories (CategoryID) Products (SupplierID) – Suppliers(SupplierID)	

**Tabla: *Region* (Descripción: región)**

Columna	Tipo de dato	Descripción
RegionID	entero	identificador de región
RegionDescription	texto	descripción de región
<b>Llave primaria:</b>	RegionID	

**Tabla: *Shippers* (Descripción: fletadores)**

Columna	Tipo de dato	Descripción
ShipperID	entero	identificador de fletador
CompanyName	texto	nombre de compañía
Phone	texto	teléfono
<b>Llave primaria:</b>	ShipperID	

**Tabla: *Suppliers* (Descripción: proveedores)**

Columna	Tipo de dato	Descripción
SupplierID	entero	identificador de proveedor
Address	texto	dirección
City	texto	ciudad
CompanyName	texto	proveedores
ContactName	texto	nombre de contacto
ContactTitle	texto	puesto de contacto
Country	texto	país
Fax	texto	fax

HomePage	texto	página
Phone	texto	teléfono
PostalCode	texto	código postal
Region	texto	región
<b>Llave primaria:</b>	SupplierID	

**Tabla: *Territories* (Descripción: territorios)**

<b>Columna</b>	<b>Tipo de dato</b>	<b>Descripción</b>
TerritoryID	texto	identificador de territorio
RegionID	entero	identificador de región
TerritoryDescription	texto	descripción de territorio
<b>Llave primaria:</b>	TerritoryID	
<b>Llave foránea:</b>	Territories (RegionID) – Region (RegionID)	

## ANEXO D. DESCRIPCIÓN DE LA BASE DE DATOS GEOBASE

Geobase es una base de datos con información geográfica de los Estados Unidos de América. Para este trabajo, se elaboró una versión de la base de datos en Microsoft Access, a partir de su esquema original en Prolog. Esta versión contiene 11 tablas y 41 columnas. En este anexo se presenta el esquema y los detalles de las tablas de la base de datos.

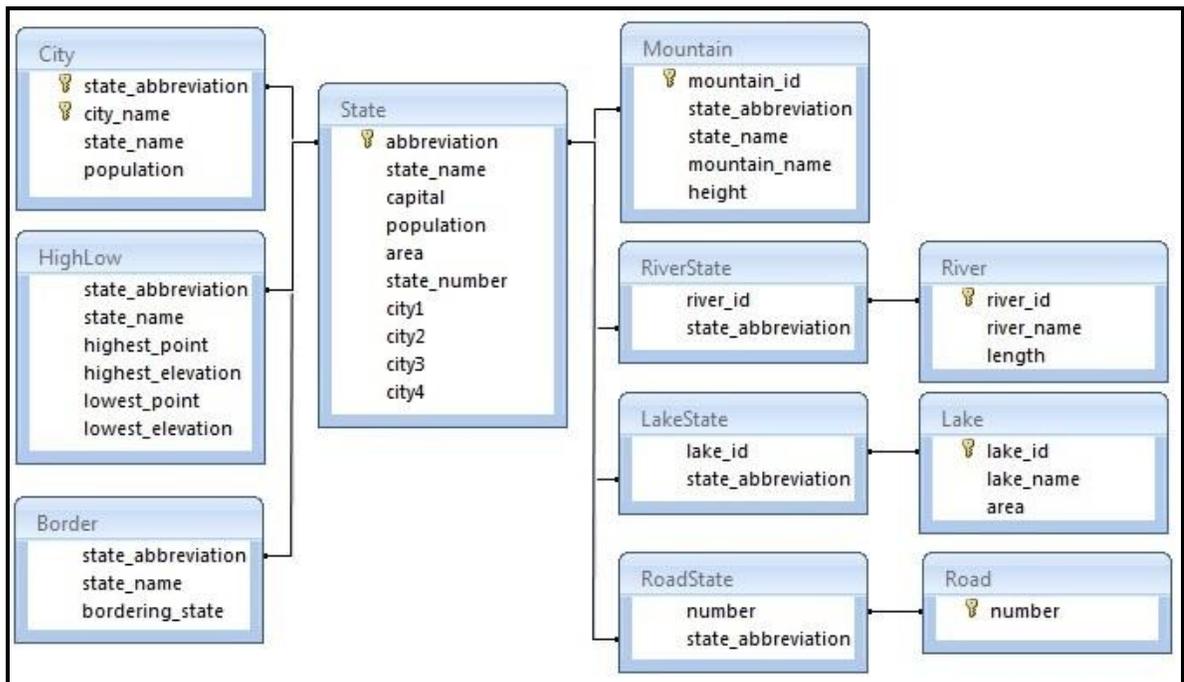


Figura D.1 Esquema de la base de datos Geobase

Tabla: **Border** (Descripción: colindancia)

Columna	Tipo de dato	Descripción
border_state	texto	nombre de estado
state_abbreviation	texto	abreviatura de estado
bordering_state	texto	colindancia
<b>Llave primaria:</b>	state_abbreviation, bordering_state	

**Llave foránea:** Border (state\_abbreviation) – State (abbreviation)

---

**Tabla: City (Descripción: ciudad)**

Columna	Tipo de dato	Descripción
state_abbreviation	texto	abreviatura de estado
city_name	texto	ciudad
population	entero	población
state_name	texto	nombre de estado
<b>Llave primaria:</b>	state_abbreviation, city_name	
<b>Llave foránea:</b>	City (state_abbreviation) – State (abbreviation)	

---

**Tabla: HighLow (Descripción: punto de estado)**

Columna	Tipo de dato	Descripción
state_abbreviation	texto	abreviatura de estado
highest_elevation	entero	elevación más alta
highest_point	texto	punto más alto
lowest_elevation	entero	elevación más baja
lowest_point	texto	punto más bajo
state_name	texto	nombre de estado
<b>Llave primaria:</b>	state_abbreviation	
<b>Llave foránea:</b>	HighLow (state_abbreviation) – State (abbreviation)	

---

**Tabla: Lake (Descripción: lago)**

Columna	Tipo de dato	Descripción
lake_id	entero	abreviatura de estado
area	entero	área
lake_name	texto	lago
<b>Llave primaria:</b>	lake_id	

---

**Tabla: LakeState (Descripción: lago estado)**

Columna	Tipo de dato	Descripción
lake_id	entero	identificador de lago

state_abbreviation	texto	abreviatura de estado
<b>Llave primaria:</b>	lake_id, state_abbreviation	
<b>Llave foránea:</b>	LakeState (lake_id) – Lake (lake_id) LakeState (state_abbreviation) – State (abbreviation)	

**Tabla: *Mountain* (Descripción: montaña)**

Columna	Tipo de dato	Descripción
mountain_id	entero	identificador de montaña
height	entero	altura
mountain_name	texto	montaña
state_abbreviation	texto	abreviatura de estado
state_name	texto	nombre de estado
<b>Llave primaria:</b>	mountain_id	
<b>Llave foránea:</b>	Mountain (state_abbreviation) – State (abbreviation)	

**Tabla: *River* (Descripción: río)**

Columna	Tipo de dato	Descripción
river_id	entero	identificador de río
length	entero	longitud
river_name	texto	río
<b>Llave primaria:</b>	river_id	

**Tabla: *RiverState* (Descripción: río estado)**

Columna	Tipo de dato	Descripción
river_id	entero	identificador de río
state_abbreviation	texto	abreviatura de estado
<b>Llave primaria:</b>	river_id, state_abbreviation	
<b>Llave foránea:</b>	RiverState (river_id) – River (river_id) RiverState (state_abbreviation) – State (abbreviation)	

**Tabla: *Road* (Descripción: carretera)**

Columna	Tipo de dato	Descripción
---------	--------------	-------------

number	entero	carretera
<b>Llave primaria:</b> number		

**Tabla: *RoadState* (Descripción: carretera estado)**

Columna	Tipo de dato	Descripción
number	entero	identificador de carretera
state_abbreviation	texto	abreviatura de estado
<b>Llave primaria:</b>	number, state_abbreviation	
<b>Llave foránea:</b>	RoadState (number) – Road (number) RoadState (state_abbreviation) – State (abbreviation)	

**Tabla: *State* (Descripción: estado)**

Columna	Tipo de dato	Descripción
abbreviation	texto	abreviatura
area	entero	área
capital	texto	capital
city1	texto	ciudad uno
city2	texto	ciudad dos
city3	texto	ciudad tres
city4	texto	ciudad cuatro
population	entero	población
state_name	texto	estado
state_number	entero	número de estado
<b>Llave primaria:</b>	abbreviation	

## GLOSARIO

**Álgebra relacional:** Es un conjunto de operadores que son utilizados para el manejo de datos en el modelo relacional; estos operadores son: unión, intersección, diferencia, producto cartesiano, selección, proyección, conjunción y división.

**Cálculo relacional:** Este cálculo se compone por el cálculo relacional de tupla y el cálculo relacional de dominio. El más conocido es el cálculo relacional de tupla, donde las consultas son fórmulas que definen conjuntos utilizando:

1. Constantes
2. Predicados (como *select* del álgebra)
3. Lógica (conjunción y, disyunción o, negación)
4.  $\exists$  (existe)
5.  $\forall$  (para todo)

**Elipsis:** Es la omisión de una o más palabras que son entendidas de manera obvia (ver sección 2.4).

**Independencia de dominio:** Característica aplicable a las interfaces de lenguaje natural para bases de datos, que se refiere a la capacidad de una interfaz para ser utilizada en la consulta a diferentes bases de datos.

## REFERENCIAS

- [Aguirre, 2014] M.A. Aguirre L., *Modelo Semánticamente Enriquecido de Bases de Datos para su Explotación por Interfaces de Lenguaje Natural*, tesis de doctorado, División de Estudios de Postgrado e Investigación, Instituto Tecnológico de Ciudad Madero, Cd. Madero, Tam., 2014.
- [Alshawi, 1994] H. Alshawi, D. Carter, R. Crouch, S. Pulman, M. Rayner, A. Smith, *CLARE: A Contextual Reasoning and Cooperative Response Framework for the Core Language Engine*, Report CRC-028, Cambridge, U.K., 1994.
- [Androutsopoulos, 1995] I. Androutsopoulos, G. Ritchie, P. Thanisch, "Natural Language Interface to Database: An Introduction", *Journal of Natural Language Engineering*, pp. 29-81, 1995.
- [Bhootra, 2004] R.A. Bhootra, *Natural Language Interfaces: Comparing English Language Front End and English Query*, tesis de maestría, Virginia Commonwealth University, 2004.
- [Boldasov, 2003] M. Boldasov, G. Sokolova, "QGen – Generation Module for the Register Restricted InBASE System", *Proc. 4th International Conference on Computational Linguistics and Intelligent Text Processing*, vol. 2588, pp. 465-476, 2003.

- [Cimiano, 2007] P. Cimiano, P. Haase, J. Heizmann, "Porting Natural Language Interfaces Between Domains: An Experimental User Study with the ORAKEL System", *Proc. 12th International Conference on Intelligent User Interfaces*, pp. 180-190, 2007.
- [ELF, 2002] ELF Software. "Access ELF FAQ (Frequently Asked Questions)", <http://www.elfsoft.com/help/accelf/HowDoI.htm>.
- [Kate, 2006] R.J. Kate, R.J. Mooney, "Using String-Kernels for Learning Semantic Parsers", *Proc. 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pp. 913-920, 2006.
- [Liddy, 1998] E. Liddy, "Natural Language Processing for Information Retrieval and Knowledge Discovery", *Visualizing Subject Access for 21st Century Information Resources*, pp. 137-147, 1998.
- [McShane, 2004] M. McShane, S. Beale, S. Niremburg, "OntoSem Methods for Processing Semantic Ellipsis", *Proc. 4th HLT-NAACL Workshop on Computational Lexical Semantics*, pp. 1-8, 2004.
- [Mellado, 2014] O.M. Mellado C. *Implementación de un Analizador Sintáctico del Idioma Español para una Interfaz de Lenguaje Natural a Bases de Datos*, tesis de maestría, División de Estudios de Postgrado e Investigación,

Instituto Tecnológico de Ciudad Madero, Cd. Madero, Tam., 2014.

- [Minock, 2007] M. Minock, "A STEP Towards Realizing Codd's Vision of Rendezvous with the Casual User", *Proc. 33rd International Conference on Very Large Data Bases*, pp. 1358-1361, 2007.
- [Minock, 2010] M. Minock, "C-Phrase: A System for Building Robust Natural Language Interfaces to Databases", *Journal of Data and Knowledge Engineering*, vol. 69, pp. 290-302, 2010.
- [Niesler, 2001] R. Niesler, C. Rous, "Natural Language Understanding in the DACST-AST Dialogue System", *Proc. 12th Annual Symposium of the Pattern Recognition Association of South Africa*, pp 134-137, 2001.
- [Nihalani, 2011] N. Nihalani, S. Silakari, M. Motwani, "Natural Language Interface for Database – A Brief Review", *International Journal of Computer Science*, vol. 8, 2011.
- [RAE, 2014] Real Academia Española, "Diccionario de la Lengua Española", <http://lema.rae.es/drae/?val=consulta>
- [Rohit, 2006] K. Rohit, R. Mooney, "Using String-Kernels for Learning Semantic Parsers", *COLING/ACL-2006*, pp. 913-920, 2006.
- [Rojas, 2009] C. Rojas, *Administrador de Diálogo para una Interfaz de Lenguaje Natural a Bases de Datos*, tesis de doctorado,

Departamento de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 2009.

- [Smith, 2001] N.A. Smith, "Ellipsis Happens, and Deletion Is How", *University of Maryland Working Papers in Linguistics*, vol. 11, 176-191. 2001.
- [Silberschatz, 2006] A. Silberschatz, H. Korth, S. Sudarshan, *Fundamentos de Bases de Datos*, 5ta edición, McGraw - Hill, 2006.
- [UPenn, 1990] University of Pennsylvania, "DARPA Air Travel Information System (ATIS0)", <https://catalog.ldc.upenn.edu/docs/LDC93S4B/sdtd.readme.html>, 1990.
- [UTexas, 1995] University of Texas, "Geoquery data", <http://www.cs.utexas.edu/users/ml/nldata/geoquery.html>, 1995.
- [Webster, 2003] N. Webster, *Merriam-Webster's Collegiate Dictionary*, 11a. edición, Britannica, 2003.
- [Zhang, 1999] G. Zhang, W. Chu, F. Meng, G. Kong, "Query Formulation from High-Level Concepts for Relational Databases", *User Interfaces to Data Intensive Systems*, p.64-74, 1999.