EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

"POR MI PATRIA Y POR MI BIEN"

TESIS

MÓDULO DE ACLARACIÓN DE CONSULTAS PARA UNA INTERFAZ DE LENGUAJE NATURAL A BASES DE DATOS

Que para obtener el Grado de
Maestra en Ciencias de la Computación

Presenta
Ing. Sandra González De La Cruz
G10070520

Director de Tesis
Dr. José Antonio Martínez Flores

Co-director de Tesis
Dr. Rodolfo A. Pazos Rangel

Cd. Madero, Tamaulipas Mayo, 2021

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Ciudad Madero
Subdirección Académica
División de Estudios de Posgrado e Investigación

Cd. Madero, Tam. 28 de mayo de 2021

OFICIO No. : U.039/21
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

C. SANDRA GONZÁLEZ DE LA CRUZ
No. DE CONTROL G10070520
P R E S E N T E

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

"MÓDULO DE ACLARACIÓN DE CONSULTAS PARA UNA INTERFAZ DE LENGUAJE NATURAL A BASES DE DATOS"

El Jurado está integrado por los siguientes catedráticos:

PRESIDENTE:	DR.	MARCO AGUIRRE LAM
SECRETARIO:	DR.	JUAN FRAUSTO SOLÍS
VOCAL:	DR.	JOSÉ ANTONIO MARTÍNEZ FLORES
SUPLENTE:	DR.	RODOLFO ABRAHAM PAZOS RANGEL
DIRECTOR DE TESIS:	DR.	JOSÉ ANTONIO MARTÍNEZ FLORES
CO-DIRECTOR DE TESIS:	DR.	RODOLFO ABRAHAM PAZOS RANGEL

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.

A T E N T A M E N T E
Excelencia en Educación Tecnológica.
"Por mi patria y por mi bien".

EDUCACIÓN
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

MARCO ANTONIO CORONEL GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

c.c.p.- Archivo
MACG 'mdcoa'

Contenido

Capítulo 1 Introducción....9
1.1 Objetivos....10
1.2 Justificación y beneficios....10
1.3 Descripción del Problema....11
1.4 Alcances y Limitaciones....13
Capítulo 2 Marco teórico y trabajos relacionados....14
2.1 Marco teórico....14
2.1.1 Procesamiento de lenguaje natural....14
2.1.2 Componentes del procesamiento del lenguaje natural....15
2.1.3 Paráfrasis....15
2.1.4 Base de datos....16
2.1.5 Sistema administrador de bases de datos (SABD)....16
2.1.6 Structured query language (SQL)....16
2.1.7 Interfaces de lenguaje natural....16
2.1.8 Interfaz de lenguaje natural a bases de datos....17
2.1.9 Ventajas y desventajas de las ILNBDs....18
2.2 Trabajos relacionados....19
2.2.1 QuestIO - Interfaz basada en preguntas para ontologías....20
2.2.2 FREyA (retroalimentación, refinamiento, agregación de vocabulario extendido)....20
2.2.3 Análisis semántico basado en reglas para ILNBD....22
2.2.4 StartMobile....22
2.2.5 Finance ontology....25
2.2.6 Aneesah....28
2.2.7 Conclusiones sobre trabajos relacionados....30
Capítulo 3 Análisis y solución conceptual del problema....32
3.1 Análisis de la interfaz de lenguaje natural para bases de datos....32
3.2 Descripción de la arquitectura actual de la ILNBD....33
3.3 Descripción de la nueva arquitectura de la ILNBD....34
3.4 Ampliación del diccionario de información semántica (DIS)....34

3.5 Diseño conceptual de la paráfrasis 36
3.6 Diseño conceptual del administrador de diálogo 38
Capítulo 4 Desarrollo del módulo de aclaración 40
4.1 Implementación de la paráfrasis 40
4.1.1 Paráfrasis del comando Select 42
4.1.2 Paráfrasis de la cláusula Where 44
4.2 Implementación del administrador de diálogo 46
4.2.1 Clase del administrador de diálogo 46
4.2.2 Validación del apartado *Muestra* 48
4.2.3 Validación del apartado *Tal que* 50
Capítulo 5 Pruebas de la ILNBD 59
5.1 Descripción del hardware y software del equipo 59
5.2 Pruebas funcionales de la ILNBD 59
5.3 Resultados 62
Capítulo 6 Conclusiones 64
6.1 Conclusiones 64
6.2 Trabajos futuros 65
Apéndice 66
Apéndice A. Corpus de consultas para la BD ATIS. 66
Apéndice B. Corpus de consultas Geoquery250 para la BD Geobase. 68
Referencias 75

Índice de Figuras

Figura 2.1 Arquitectura general de una ILN. 17
Figura 2.2 Arquitectura general de una ILNBD 18
Figura 2.3 Prototipo QuestIO. 20
Figura 2.4 Cuadro de diálogo de aclaración. 21
Figura 2.5 Interacción de solicitud-respuesta con START. 23
Figura 2.6 Sistema StartMobile. 25
Figura 2.7 Arquitectura del sistema. 26
Figura 2.8 Interacción con la interfaz de diálogo resaltando ambigüedad. 28
Figura 2.9 Arquitectura de la ILNBD Aneesah. 29
Figura 2.10 Interfaz de usuario Aneesah. 30
Figura 3.1 Arquitectura de la ILNBD versión para web [González, 2018]. 33
Figura 3.2 Nueva arquitectura de la interfaz de consulta. 34
Figura 3.3 Diseño conceptual de la paráfrasis. 36
Figura 3.4 Diseño de la consulta SQL parafraseada. 37
Figura 3.5 Diseño del administrador de diálogo. 38
Figura 3.6 Esquema de la interpretación final y resultados. 39
Figura 4.1 Diseño de la interpretación de la interfaz. 40
Figura 4.2 Paráfrasis del comando SELECT. 43
Figura 4.3 Paráfrasis de la cláusula WHERE. 45
Figura 4.4. Diálogo de aclaración: Debe ingresar al menos un elemento. 48
Figura 4.5. Diálogo de aclaración: Específica el elemento del apartado *Muestra* 49
Figura 4.6. Diálogo de aclaración: No se encontró el elemento del apartado *Muestra* 49
Figura 4.7. Diálogo de aclaración: Específica el elemento del apartado *Tal que*. 52
Figura 4.8. Diálogo de aclaración: No se encontró el elemento del apartado *Tal que* 53
Figura 4.9. Diálogo de aclaración: Especifica la comparación. 54
Figura 4.10. Diálogo de aclaración: Especifica el rango de condición. 55
Figura 4.11. Diálogo de aclaración: No se encontró la comparación de la condición. 56
Figura 4.12. Diálogo de aclaración: No se encontró el valor de búsqueda. 56
Figura 4.13. Interpretación y resultados de la consulta. 58

Índice de Tablas

Tabla 2.1 Tabla descriptiva de interfaces con diálogo para aclaración de consultas. 30
Tabla 3.1 Estructura de la tabla *palabras_reservadas*. 35
Tabla 5.1 Características del hardware. 59
Tabla 5.2 Características del software. 59
Tabla 5.3 Resultados obtenidos con el corpus de la base de datos ATIS... 63
Tabla 5.4 Resultados obtenidos con el corpus de la base de datos Geobase. 63

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 en 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 las publicaciones.

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

_31_ de _____Mayo____ de __2021__, Cd. Madero, Tamps.

Ing. Sandra González de la Cruz

Agradecimientos

Agradezco a los miembros del Comité Tutorial de esta tesis, conformado por el Dr. José A. Martínez, Dr. Rodolfo A. Pazos, Dr. Juan Frausto Solís y Dr. Marco A. Aguirre por sus observaciones y sugerencias brindadas durante el desarrollo de esta tesis.

Particularmente agradezco a mi Asesor de Tesis el Dr. José A. Martínez por dar seguimiento atento a este proyecto y además haber aportado sus conocimientos, apoyo y tiempo para el desarrollo de este trabajo, a mi Co-director de Tesis el Dr. Rodolfo A. Pazos quien también aportó de sus conocimientos, experiencia y apoyo en el desarrollo de este trabajo.

Gracias al Instituto Tecnológico de Cd. Madero (ITCM) por brindarme la oportunidad de continuar con mis estudios, y al Consejo Nacional de Ciencia y Tecnología (CONACYT) por haberme otorgado una beca para el desarrollo de esta investigación.

Muchas gracias a todos mis compañeros de generación por su apoyo, confianza y todas las experiencias que compartimos juntos durante la Maestría, principalmente a Jessica E. González San Martín por brindarme su apoyo y amistad.

Agradezco a Dios por permitirme llegar a esta etapa de mi vida y poder realizar mis estudios, y a mi familia, por todo su apoyo incondicional.

Dedicatoria

Dedico esta tesis a mis padres Alfonso González Josefa y Margarita de la Cruz Cruz por el amor, la confianza y el apoyo que siempre me han brindado en los momentos más decisivos de mi vida. Gracias a sus consejos y motivación he podido cumplir cada una de mis metas.

A mis hermanos Jesús Daniel, José Luis y Andrea González de la Cruz por el cariño incondicional que nos témenos y el apoyo que siempre me han brindado.

Resumen

En las últimas décadas, el uso de la tecnología ha avanzado constantemente, por la facilidad y rapidez de repuesta que éstas pueden brindar a los usuarios y por lo que hoy en día juega un rol importante en nuestra vida cotidiana, esto ha llevado a muchos empresarios a actualizarse y a generar un enorme almacenamiento de información en Bases de Datos (BDs), la cual es consultada para importantes tomas de decisiones.

Cabe mencionar que para obtener información de una base de datos se requiere formular consultas en lenguaje formal de modo que la computadora sea capaz de interpretar, por lo que un usuario que carece de este conocimiento no le sería una tarea fácil de realizar y le tomaría tiempo. Por lo cual, muchos de los investigadores se han dado a la tarea de crear herramientas con una interfaz que permita la interacción entre usuario-máquina de una forma fácil y natural como son las Interfaces de Lenguaje Natural a Bases de Datos (ILNBDs).

Las ILNBDs facilitan a usuarios comunes realizar consultas a BDs para obtener información, escribiendo éstas en Lenguaje Natural (LN), ya que las interfaces la traducen a un lenguaje formal como el SQL (por sus siglas en inglés, Structured Query Language) de manera automática. Uno de los problemas surge cuando las consultas en LN no están correctamente formuladas por los usuarios, y por lo tanto las ILNBDs no realizan la interpretación de forma correcta.

Con el fin de mejorar la interpretación de las consultas, este proyecto presenta la ampliación de la funcionalidad de una ILNBD para web desarrollada en el ITCM, que consiste en el desarrollo de un módulo para la aclaración de consultas en LN en el idioma español, en el cual, a partir de la consulta SQL obtenida del traductor de la interfaz, el usuario puede visualizar previamente mediante una paráfrasis la consulta y en caso de ser necesario puede refinarla con un administrador de diálogo para obtener los resultados deseados.

El administrador de diálogo cuenta con tres opciones y un proceso de diálogos que ayudan al usuario a clarificar la consulta en caso de ser necesario para que esta pueda ser procesada. En la experimentación realizada a este módulo se utilizaron dos subconjuntos de corpus de BDs (ATIS y Geobase) donde se obtuvo entre el 92-94% de eficiencia en la interfaz.

Capítulo 1
Introducción

La integración de una Base de Datos (BD) en aplicaciones es muy común, con el tiempo se almacena una enorme cantidad de información en éstas, que en su mayoría se utiliza para el análisis y mejora de negocios, así como para el servicio al cliente. Sin embargo, quienes administran y acceden a información de BDs deben ser usuarios con conocimientos en un lenguaje de consulta como SQL (Structured Query Language).

Actualmente existen herramientas como son las Interfaces de Lenguaje Natural a Bases de Datos (ILNBDs) que facilitan a usuarios comunes realizar consultas a BDs, escribiendo éstas en Lenguaje Natural (LN). Es decir, con ILNBDs el usuario para obtener información de BDs no requiere de un conocimiento previo o experiencia con un lenguaje de consulta a BDs, ya que éstas traducen de LN al lenguaje SQL de manera automática.

En 2013 la Revista Internacional de Ciencia y Tecnología Avanzada menciona un sistema que apareció a finales de los sesenta (C. Rendezvous System), este sistema permite a los usuarios acceder a BDs a través de un LN relativamente ilimitado. En este sistema se hace especial hincapié en la paráfrasis de consultas y en involucrar a los usuarios en diálogos de aclaración cuando hay dificultades para analizar la entrada del usuario.

En 2014 se concluyó la tesis de doctorado "Modelo Semánticamente Enriquecido de Bases de Datos para su Explotación por Interfaces de Lenguaje Natural". El propósito de dicha tesis consistió en desarrollar una ILNBD independiente de dominio (aplicable a cualquier BD), la cual permite traducir consultas en idioma español a SQL. Es importante destacar que la ILNBD con el corpus ATIS obtiene un desempeño de 90% de consultas contestadas correctamente, lo cual la hace competitiva con las mejores ILNBDs independientes de dominio desarrolladas por otros investigadores.

Este proyecto consiste en ampliar la funcionalidad de una ILNBD que le permita aclarar consultas en LN formuladas por usuarios, con el propósito de mejorar la interpretación de la interfaz; para lo cual en este Capítulo se describe el objetivo general y específicos del proyecto, la justificación y beneficios que se obtendrán, el marco teórico, descripción del problema, metodología de la solución, alcances y limitaciones del proyecto.

1.1 Objetivos

Objetivo general:

Ampliar la funcionalidad de una interfaz de lenguaje natural a bases de datos para que permita aclarar consultas formuladas por los usuarios en lenguaje natural en el idioma español con el fin de mejorar la interpretación de éstas por la interfaz.

Objetivos específicos:

OE.1) Analizar la estructura general de consultas en SQL (instrucción SELECT), con el fin de definir una estructura de la consulta en lenguaje natural que se genere a partir de una instrucción en SQL.

OE.2) Definir una solución conceptual para generar una paráfrasis de una consulta en LN a partir de una consulta en SQL, y el estudio de una posible ampliación del Diccionario de Información Semántica de la ILNBD.

OE.3) Implementar el algoritmo para generar la paráfrasis.

OE.4) Diseñar e implementar un administrador de diálogo que permita aclarar la consulta original (por ejemplo, tablas y columnas).

1.2 Justificación y beneficios

Existe una enorme cantidad de información en diversas BDs que día a día es actualizada, a la cual solamente usuarios expertos en la materia pueden acceder mediante un lenguaje formal, sin embargo, en la actualidad mediante una ILNBD es posible traducir el LN que usa el ser humano a un lenguaje SQL para poder acceder a la información.

Puede ocurrir ocasionalmente que las ILNBDs no interpreten las consultas de forma correcta, por ejemplo:

- Un caso podría ser porque no están correctamente formuladas en LN por los usuarios, una manera de evitar que esto ocurra es notificar al usuario cómo interpretó la consulta la interfaz, además de preguntar al usuario si la interpretación es correcta, en caso de que no sea la interpretación correcta, se debe generar un diálogo con el usuario para aclarar la consulta. Por ejemplo, el caso descrito en [Sujatha, 2012], acerca de un sistema TQA que contiene un módulo de paráfrasis que convierte la consulta SQL nuevamente a LN. La solicitud en LN reconstruida se presenta al usuario, para garantizar que ninguna de las etapas intermedias de transformación haya causado que su solicitud sea malinterpretada.

- Otro caso que podría acontecer es por fallas propias del desarrollo de las ILNBDs.

Por otra parte, muchos de los investigadores que han desarrollado sistemas de ILNBDs se han enfocado en el desempeño y eficiencia de resultados certeros y confiables en lugar de realizar un diálogo de aclaración con el usuario para obtener el resultado deseado.

De este modo surge la necesidad de ampliar la funcionalidad de la ILNBD que se desarrolla en el ITCM para mejorar la interpretación de lo que los usuarios consultan; así también, se pretende aumentar el porcentaje de respuestas certeras y confiables, reduciendo el margen de error y evitando que el usuario tenga que aprender un lenguaje formal como el SQL.

1.3 Descripción del Problema

Las ILNBDs son herramientas que han sido de gran utilidad para usuarios casuales debido a la facilidad de poder consultar y obtener información almacenada en una BD mediante peticiones escritas en un LN sin la necesidad de dominar un lenguaje de consulta a BDs. Sin embargo, muchos de los proyectos de ILNBDs desarrollados por los investigadores no han logrado un desempeño adecuado (cerca del 100% de consultas contestadas correctamente).

Para garantizar los mejores resultados de la interfaz, se requiere que ésta parafrasee la instrucción SQL que generó, a una oración en LN, antes de enviar la instrucción SQL al SABD para extraer la información de la BD, esto con el fin de verificar con el usuario que el sistema ha procesado de manera correcta la petición.

Cabe mencionar que la paráfrasis es un proceso muy difícil, esto sucede por razones variadas, principalmente por que el español es uno de los lenguajes de mayor complejidad, debido al alto grado de libertad que se tiene para la redacción de oraciones [Mellado, 2014].

Este trabajo se centra en aclaración de consultas para una ILNBD, desarrollada para los usuarios que carecen de conocimientos de BDs y sobre todo de un lenguaje de consulta a éstas.

Por otra parte, durante el procesamiento de lenguaje natural (PLN) pueden ocurrir problemas de traducción. Por lo tanto, de acuerdo a [Pazos, 2013] en un análisis llevado a cabo en corpus de consultas que involucran tres bases de datos (ATIS, Northwind y Pubs), se identificaron y clasificaron cuatro tipos generales de problemas que generalmente se encuentran en la mayoría de las consultas a bases de datos:

1) Las diferentes categorías sintácticas de las palabras o sintagmas que se pueden usar para referirse a tablas y columnas de la BD:

   - Sustantivos

   Ejemplo: *Enumere el número de asientos en D9S.*

- Verbos
  Ejemplo: *¿A qué hora sale el vuelo 102136 de ATL a DFW?*

- Adjetivos
  Ejemplo: *¿Qué tan rápido puede volar el Concorde?*

- Preposiciones
  Ejemplo: *Dame un vuelo en clase económica de DFW a BWI de ida.*

2) La elipsis semántica, que ocurre cuando se omiten palabras que son necesarias para el claro entendimiento de la consulta.

   Ejemplo: *¿Cuánto cuesta el vuelo 539 de Delta?*

3) La cobertura de la capacidad de SQL, como las consultas que involucran varias tablas.

   Ejemplo: *Dame un vuelo en clase económica de DFW a BWI de ida.*

4) Otro tipo de problemas que involucran errores humanos, información inexistente en la BD, palabras que indican valores imprecisos y alias.

   Errores de espaciado, puntuación y formato.
   Ejemplo: *vuelos entre SFO y Dallas entre el mediodía y las 5:00 p.m.*

   Valores de búsqueda imprecisos.
   Ejemplo: *muéstrame los vuelos de Atlanta a Dallas por la mañana.*

   Valores de búsqueda de alias.
   Ejemplo: *muéstrame los vuelos que salen después del mediodía.*

Por lo anterior, se puede decir que la tarea de una ILNBD no es un proceso fácil de realizar, ya que ésta suele ser compleja, confusa y requiere una profunda comprensión del funcionamiento de la interfaz para un resultado exitoso.

1.4 Alcances y Limitaciones

Alcances:

- La aclaración de consultas es mediante un lenguaje natural, la cual es en el idioma español, mismo idioma que se genera la consulta original en el módulo de ILNBD, esto para facilitar la interacción con el usuario.

- El módulo muestra mediante una paráfrasis lo que la interfaz interpretó, confirmando con el usuario la solicitud que requirió al inicio, para la cual el usuario tiene la opción de continuar con el proceso si ésta fue interpretada de manera correcta, en caso contrario, puede reformular su solicitud con una idea más clara después de haber visto la paráfrasis.

- Con la retroalimentación del usuario se puede eliminar el mayor número de problemas que generalmente se encuentran en las consultas para la mayoría de las bases de datos, mencionadas anteriormente en la descripción del problema.

- Se implementó un nuevo módulo que evite al máximo la modificación del código de la interfaz.

Limitaciones:

- El idioma a tratar en el proceso de paráfrasis es únicamente en español.

- No se proporciona información que no se encuentre explícitamente en la BD, es decir, no se tratarán consultas de bases de datos deductivas.

- No se interpretan consultas que involucren funciones de agregación, agrupación y de orden.

- No se realizaron las pruebas de usabilidad de la interfaz con usuarios de prueba. El motivo es que se requiere de la participación de usuarios de prueba, los cuales no siempre están disponibles cuando se requieren. Además, el éxito de las pruebas depende de factores incontrolables, por ejemplo, un experimento puede resultar inservible si ocurre un apagón durante las pruebas, ya que los usuarios de prueba ya no podrían repetir el experimento, debido a que en la segunda ocasión ya tendrían experiencia en el uso de la interfaz, y los resultados estarían sesgados.

Capítulo 2
Marco teórico y trabajos relacionados

En este capítulo se presenta el marco teórico y los trabajos relacionados que conforma este proyecto de tesis. En el marco teórico se presentan las definiciones más importantes para una mejor comprensión del presente proyecto. En los trabajos relacionados se describen los trabajos que se investigaron durante la realización de este proyecto de tesis.

2.1 Marco teórico

En esta sección se describen los conceptos básicos utilizados durante el desarrollo de este proyecto para el conocimiento sobre el tema del proyecto.

2.1.1 Procesamiento de lenguaje natural

Se describen los siguientes conceptos de lenguaje descritos en [Rojas, 2009]:

Lenguaje: Es la función que permite expresar pensamientos y comunicaciones entre las personas para lograr el entendimiento con otras, mediante un conjunto de oraciones formuladas por sonidos o símbolos. Se distingue entre dos clases, los lenguajes naturales como el castellano o el inglés, y los lenguajes formales como las matemáticas y la lógica.

Lenguaje Natural: Es el medio de comunicación que el hombre utiliza de modo cotidiano para establecer una relación con los demás.

Sin embargo, es necesario que las combinaciones de palabras utilizadas para crear una oración sean correctas con respecto a una sintaxis y tengan sentido con respecto a la semántica, debido a que no todas las combinaciones de palabras son permitidas.

Lenguaje Formal: Es un lenguaje artificial que el hombre ha desarrollado, compuesto por símbolos y fórmulas para expresar simbólicamente y formalizar el conocimiento científico de cada área. Las palabras y oraciones de un lenguaje formal son perfectamente definidas, es decir, una palabra mantiene el mismo significado prescindiendo de su contexto o uso.

Procesamiento de lenguaje natural (PLN): Se puede considerar un sistema inteligente cuando éste es capaz de interactuar con un usuario en su mismo leguaje, por

ejemplo, existen sistemas de Procesamiento de Lenguaje Natural (PLN) que reciben como entrada el LN o común del ser humano y generan el mismo o diferente lenguaje de salida.

Conceptualmente el procesamiento del LN es un conjunto de técnicas computacionales para analizar y representar textos de forma natural en uno o más niveles de análisis lingüístico, para llevar a cabo el procesamiento del lenguaje como un ser humano para una serie de tareas y aplicaciones [Liddy, 1998].

2.1.2 Componentes del procesamiento del lenguaje natural

A continuación, se presentan algunos de los componentes del PLN. La aplicación de estos análisis depende del objetivo de la tarea a realizar del PLN.

Análisis morfológico o léxico: Consiste en el análisis interno de las palabras que forman oraciones para extraer lemas, rasgos flexivos, unidades léxicas compuestas. Es esencial para la información básica: categoría sintáctica y significado léxico.

Análisis sintáctico: Consiste en el análisis de la estructura de las oraciones de acuerdo con el modelo gramatical empleado (lógico o estadístico).

Análisis semántico: Proporciona la interpretación de las oraciones, una vez eliminadas las ambigüedades morfosintácticas.

Análisis pragmático: Incorpora el análisis del contexto de uso a la interpretación final. Aquí se incluye el tratamiento del lenguaje figurado (metáfora e ironía) así como el conocimiento del mundo específico necesario para entender un texto especializado.

Un sistema de PLN consiste de los siguientes procesos:

- El usuario le expresa (de alguna forma) a la computadora qué tipo de procesamiento desea hacer.
- La computadora analiza las oraciones proporcionadas, en el sentido morfológico y sintáctico.
- Luego, se analizan las oraciones semánticamente, es decir, se determina el significado de cada oración.
- Se realiza el análisis pragmático del texto. Así, se obtiene una expresión final.

2.1.3 Paráfrasis

El diccionario de la Real Academia Española (RAE) señala que la paráfrasis es "la explicación o interpretación amplificativa de un texto para ilustrarlo o hacerlo más claro o

inteligible", así también como “Frase que, imitando en su estructura otra conocida, se formula con palabras diferentes” [RAE, 2020].

2.1.4 Base de datos

Un sistema de base de datos (BDs) es básicamente un sistema computarizado para guardar registros; es decir, es un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones [Faudón, 2001].

2.1.5 Sistema administrador de bases de datos (SABD)

Consiste en una colección de datos interrelacionados y un conjunto de programas para accederlos. Esta colección de datos se denomina comúnmente base de datos, y contiene información relevante para una organización. El objetivo de un SABD es proporcionar una forma de almacenar y recuperar la información de una BD de manera que sea lo más práctica y eficiente [Rojas, 2009].

2.1.6 Structured query language (SQL)

SQL (lenguaje de consultas estructurado) fue desarrollado por IBM. SQL es una combinación de constructores de álgebra relacional y del cálculo racional. Usando SQL se puede definir la estructura de los datos, además de poder modificar los datos de la base datos y especificar restricciones de seguridad. Actualmente SQL es el estándar de uso en la inmensa mayoría de los SABDs comerciales [Osorio, 2008].

2.1.7 Interfaces de lenguaje natural

Las Interfaces de Lenguaje Natural (ILN) son mecanismos que permiten a una persona interactuar con una máquina con un LN o cotidiano, por lo que el usuario no requiere de un conocimiento especializado para la interacción.

Generalmente, esta comunicación es bidireccional, es decir, del tipo de pregunta y respuesta. Las ILNs son un área de estudio activa en el campo del PLN y la lingüística computacional [Aguirre, 2014]. En la Figura 2.1 se representa una arquitectura general de una ILN.

Figura 2.1 Arquitectura general de una ILN

2.1.8 Interfaz de lenguaje natural a bases de datos

Las interfaces de lenguaje natural para bases de datos (ILNBDs) son sistemas que traducen una oración de LN a un lenguaje de consulta de BDs [Androutsopoulos, 1995]. En la Figura 2.2 se presenta una arquitectura general de una ILNBD.

En el procesamiento de una ILNBD, generalmente el resultado se presenta de dos maneras, como una respuesta en LN o un conjunto de datos generalmente en forma tabular que se devuelve al ejecutar una consulta SQL, como se muestra en el siguiente ejemplo [González,2005]:

Consulta LN: *Dame el nombre de los empleados donde la sucursal sea igual a Tampico*
Consulta SQL: SELECT Nombre, Apellido_Paterno, Apellido_Materno
FROM Empleados
WHERE Sucursal = 'Tampico'

Resultado:

Nombre	**Apellido_Paterno**	**Apellido_Materno**
Francisco	López	Mellado
Javier	González	Velazco
Teresa	Rodríguez	Leal
Alfredo	Gaytán	Álvarez

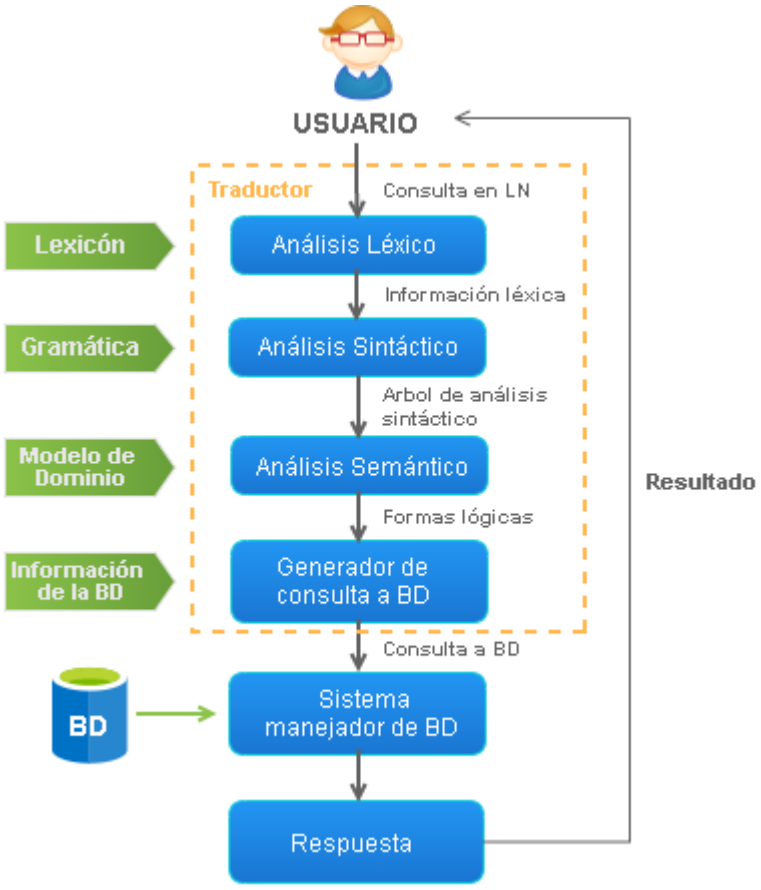


Figura 2.2 Arquitectura general de una ILNBD

2.1.9 Ventajas y desventajas de las ILNBDs

En la literatura se han mencionado diversas ventajas y desventajas de las ILNBDs [Androutsopoulos, 1995].

Entre las ventajas de este tipo de sistemas podemos mencionar:

No se requiere conocimiento de un lenguaje de consulta a BDs. El usuario no tiene la necesidad de aprender un lenguaje de consulta a BDs, el cual puede ser difícil de aprender y dominar, al menos para usuarios sin conocimientos especiales en computación.

Mejor para algunas consultas. Existen algunos tipos de preguntas que pueden ser expresadas fácilmente en LN, pero que pueden ser difícil de expresar a través de interfaces gráficas o basadas en formularios. Por ejemplo, las preguntas que incluyen negación, tal como: ¿Cuáles departamentos no tienen programadores?, o las que incluyen universalidad como: ¿Cuáles compañías suministran a todos los departamentos?

Permiten el discurso. Se mejora el diálogo al usar expresiones anafóricas mediante el uso de preguntas breves donde el significado de cada una de las preguntas es complementado por el contexto del discurso.

Algunas de las desventajas de las ILNBDs mencionadas en la literatura son las siguientes:

Cobertura lingüística no obvia. Una de las quejas frecuentes respecto a las ILNBDs es el hecho de que las capacidades lingüísticas del sistema no son obvias para el usuario [Tennant, 1983]. En otras palabras, los usuarios encuentran difícil entender y recordar qué tipos de preguntas puede o no hacer. Por ejemplo, si el sistema responde de manera correcta a un tipo de pregunta, el usuario puede suponer que el sistema es capaz de dar respuesta a todas las preguntas de ese tipo, hecho que no es cierto. Además, si el sistema provee una respuesta equivocada a un tipo de pregunta, el usuario puede suponer que todas las preguntas de ese tipo no serán respondidas por el sistema.

Fallas lingüísticas vs conceptuales. Cuando una ILNBD es incapaz de interpretar una pregunta, el usuario no sabe si esto se debe a que la pregunta está fuera de la cobertura lingüística del sistema o si está fuera de la cobertura conceptual del mismo [Tennant, 1983]. En ocasiones cuando el usuario piensa que el problema se debe a la limitada cobertura lingüística, éste intenta reformular la pregunta utilizando diferentes conceptos que el sistema desconoce. En otras ocasiones, el usuario no intenta reformular la pregunta, pues no se imagina que la pregunta formulada de esa manera no puede ser respondida por el sistema, aunque una reformulación de la misma podría ser interpretada correctamente por el sistema.

Los usuarios suponen inteligencia por parte del sistema. A menudo, los usuarios de ILNBDs suponen que el sistema es inteligente. Por ejemplo, si el sistema provee acceso a través de LN a cierta información de la BD, los usuarios tienden a creer que el sistema puede deducir otros hechos a partir de esa información, hechos que, a pesar de no estar explícitamente codificados, resultan obvios para cualquiera con sentido común [Hendrix,1982].

2.2 Trabajos relacionados

Actualmente, existe una gran variación de proyectos de ILNs creados por diversos investigadores en el mundo, que con el tiempo se han mejorado, la gran mayoría de estos trabajos están enfocados en la traducción del LN a SQL. Cabe mencionar que hoy en día son muy pocos los que cuentan con un proceso que genere una paráfrasis de una consulta en LN a partir de una consulta en SQL, la cual es el principal objetivo de este proyecto. A continuación, se describen algunos trabajos relacionados con el tema.

2.2.1 QuestIO - Interfaz basada en preguntas para ontologías

La interfaz del prototipo descrita en [Damljanovic, 2011] construye el léxico de dominio automáticamente a partir de los recursos semánticos e intenta interpretar automáticamente la consulta del usuario en función de los mecanismos de clasificación internos que se basan en el razonamiento ontológico.

Esta interfaz contiene un cuadro de texto para una consulta y un botón. Cada página siempre tiene un enlace para navegar por la ontología, de modo que los usuarios se sientan más cómodos con el formato estructurado y puedan usarlo. Después de que el usuario envía una consulta, los resultados se muestran en un panel de documentos (el panel donde los resultados son URL de documentos que mencionan conceptos de la consulta), y un panel de referencia que se utiliza para refinar el conjunto de documentos devueltos en el panel del documento, o para proporcionar una respuesta.

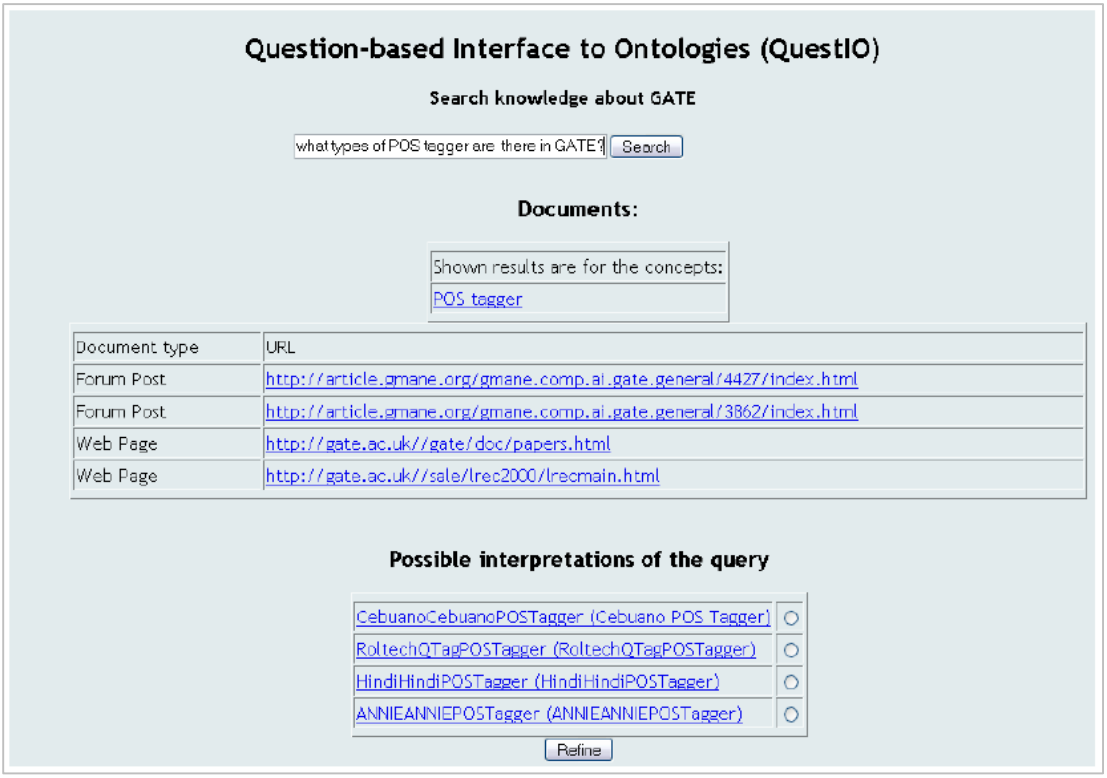


Figura 2.3 Prototipo QuestIO.

2.2.2 FREyA (retroalimentación, refinamiento, agregación de vocabulario extendido)

Otro sistema descrito en [Damljanovic, 2011] es FREyA un sistema interactivo que explora:

Retroalimentación: se refiere a mostrar las interpretaciones de las consultas del usuario. Donde el enfoque de QuestIO se extiende con un método interactivo para mostrar las interpretaciones de las consultas al usuario y permitirles elegir la correcta.

Refinamiento: se refiere a resolver ambigüedades que surgen debido a la amplia cobertura de dominio. Las ambigüedades se resuelven al involucrar al usuario con diálogos de aclaración.

El **vocabulario extendido** donde además del léxico que genera automáticamente (como en QuestIO), el vocabulario del usuario y la detección de sinónimos junto con un mecanismo de aprendizaje se utilizan para mejorar el rendimiento del sistema.

El algoritmo de consolidación tiene como objetivo fusionar el resultado de la búsqueda basada en ontología y el análisis sintáctico y análisis por mapeo de los COP (Concepto de Ontología Potencial) identificados en CO (Concepto de Ontología). Si bien este algoritmo intenta realizar este paso automáticamente, es posible que requiera atención del usuario. Este es el caso cuando hay CO ambiguos en la pregunta que no se pudieron resolver automáticamente, o cuando un COP no se pudo asignar a un CO automáticamente. Más concretamente, un COP se asigna a un CO de dos maneras: Automático y al involucrar al usuario.

Cuando el sistema no genera automáticamente la respuesta (o cuando está configurado para trabajar en el modo forceDialog), le indicará al usuario un diálogo. Hay dos tipos de diálogos en FREyA:

1. El cuadro de diálogo Desambiguación: involucra al usuario para resolver las ambigüedades identificadas en la pregunta.

2. El cuadro de diálogo Asignación: implica que el usuario asigne un COP a uno de los CO sugeridos.

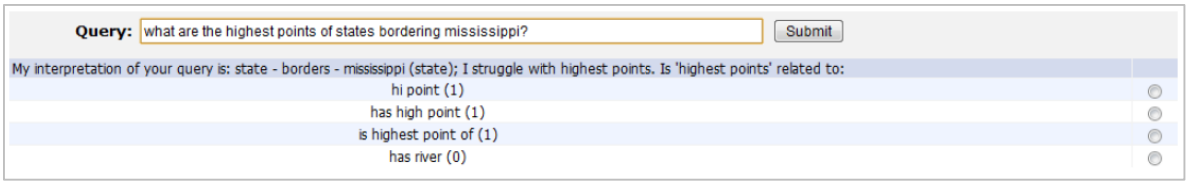


Figura 2.4 Cuadro de diálogo de aclaración.

2.2.3 Análisis semántico basado en reglas para ILNBD

El sistema de ILNBD [Anand, 2017] permite al usuario final que no esté familiarizado con las consultas SQL y no pueda escribir consultas SQL complejas pueda obtener datos utilizando el idioma inglés normal. Considere un ejemplo: "¿Quién enseña PLN?" Para la computadora, el término "enseña" aquí es ambiguo, ya que puede asociar un tema al maestro y al maestro a los estudiantes. Por el contrario, un humano sabrá de inmediato que "enseña" aquí se refiere a la relación profesor-materia porque la PLN es una "materia". Pero una persona, que no conoce la sintaxis de la base de datos TSQL, no podrá acceder a la base de datos CollegeDB a menos que conozca el SQL. Pero usando PLN, acceder a la base de datos será mucho más simple.

El sistema principal de la ILNDB incluye:

1. Interfaz de usuario: este módulo recibe información de un usuario en formato de texto o audio de voz para PLN.

2. Comprensión del lenguaje natural: basado en las reglas y los marcos semánticos, este módulo comprende el idioma inglés e identifica el campo clave para la generación de consultas

3. Generación de consulta de base de datos: después de comprensión del lenguaje natural de los campos clave identificados, este módulo convierte la entrada del usuario al lenguaje de consulta de base de datos como SQL. De la consulta SQL, los datos obtenidos se recuperan de la base de datos respectivamente.

4. Generación de lenguaje natural: para el usuario, el módulo de interfaz de usuario muestra la salida en lenguaje natural.

El sistema incluye un diálogo único y un enfoque de diálogo múltiple. Para el diálogo único, utiliza PLN que involucra consultas complejas relacionadas con la unión y múltiples declaraciones de unión. Y para los múltiples diálogos, utiliza PLN para utilizar una amplia gama de temas, como el registro del curso. Cada diálogo múltiple tiene una secuencia de interacción usuario-sistema (o turnos). En promedio, cada diálogo contiene alrededor de 2 a 4 respuestas.

2.2.4 StartMobile

La interfaz de lenguaje natural StartMobile para dispositivos móviles descrito por [Katz, 2018] utiliza el sistema START para crear este sistema, que proporciona una interfaz de lenguaje natural para dispositivos móviles. StartMobile permite a sus usuarios realizar solicitudes en inglés de información presente en sus dispositivos móviles, emitir comandos para realizar acciones en sus dispositivos y hacer que las solicitudes de información estén

disponibles desde una amplia gama de fuentes más allá de los límites de su dispositivo. Las solicitudes pueden ingresarse por escrito o por voz, utilizando las utilidades de reconocimiento de voz que ofrece Google, Inc.

En su papel tradicional de responder preguntas, START ofrece respuestas que se basan en fuentes de información que incluyen materiales estructurados, semiestructurados y no estructurados. Algunos de estos materiales se mantienen localmente y a otros se accede de forma remota a través de Internet.

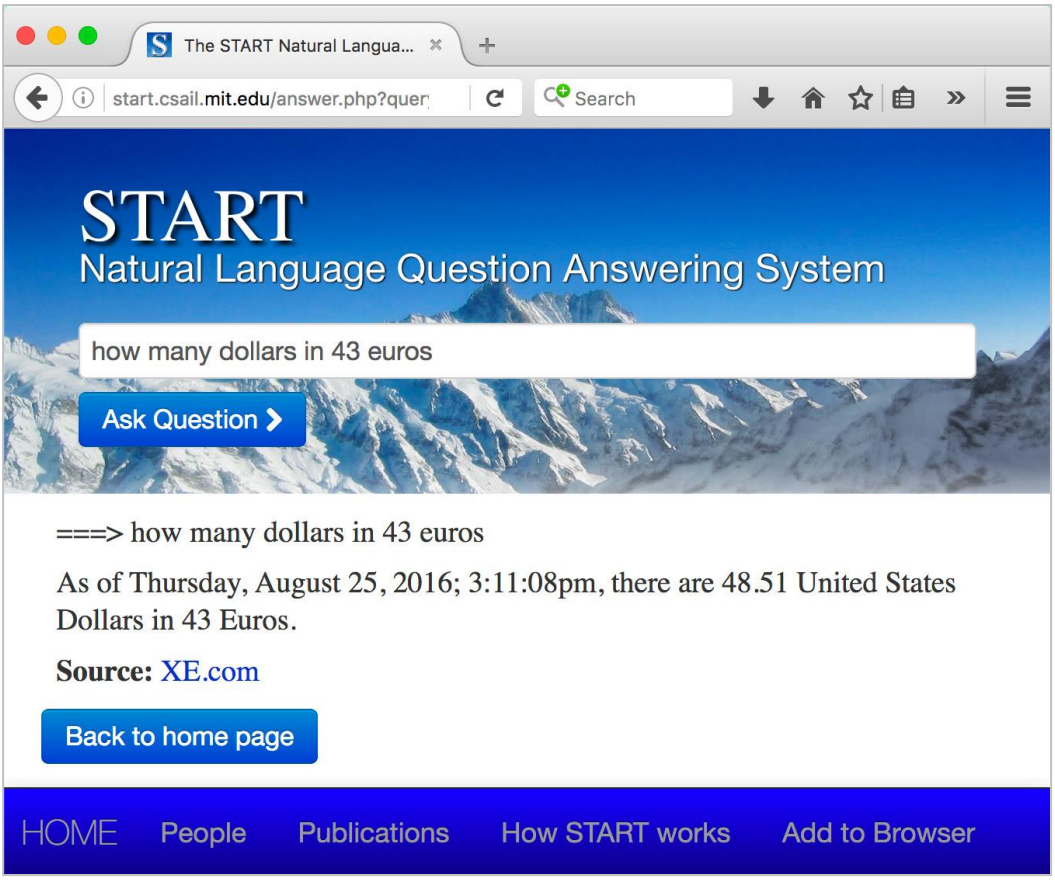


Figura 2.5 Interacción de solicitud-respuesta con START.

Un aspecto particularmente importante del diseño de START es el uso de expresiones ternarias como una representación interna de las expresiones del lenguaje natural. Las expresiones ternarias representan el lenguaje como un conjunto de triples anidados sujeto-relación-objeto, donde el sujeto y el objeto pueden ser expresiones ternarias. La representación de la expresión ternaria es una representación versátil del lenguaje basada en la sintaxis que resalta las relaciones semánticas significativas y permite la codificación detallada de las características sintácticas y léxicas.

Por ejemplo, supongamos que se presenta START con una declaración:

Grecia sorprendió a la Unión Europea con sus acciones.

Esta declaración también se puede parafrasear ya que "las acciones de Grecia sorprendieron a la Unión Europea". Para hacer coincidir las preguntas relacionadas con esta versión alternativa de la declaración, START debe hacer uso de una regla de transformación estructural que se puede expresar de la siguiente manera:

Si *<< sujeto verbo objeto1> con objeto2>*
Entonces *<objeto2 verbo objeto1> Y*
<objeto2 relacionado con el sujeto>

Donde el verbo pertenece a la clase de reacción emocional

Con la adición de esta regla, START puede responder no solo preguntas como:

¿Sorprendió Grecia a la Unión Europea con sus acciones?

¿Sorprendió Grecia a la Unión Europea?

También puede responder preguntas como:

¿Las acciones de Grecia sorprendieron a la Unión Europea?

¿Qué acciones de país sorprendieron a la Unión Europea?

START analiza estas anotaciones y almacena las estructuras analizadas (expresiones ternarias anidadas) con punteros de vuelta al segmento de información original. Para responder una pregunta, la consulta del usuario se compara con las anotaciones almacenadas en la base de conocimiento. Si se encuentra una coincidencia entre las expresiones ternarias derivadas de las anotaciones y las derivadas de la consulta, el segmento anotado correspondiente se devuelve al usuario como respuesta.

StartMobile utiliza el sistema START como primera etapa en el procesamiento de las solicitudes de los usuarios. START realiza una interpretación inicial de las solicitudes, y si estas solicitudes se refieren a la recuperación de información general de la World Wide Web u otras fuentes, START obtiene la información para presentarla al usuario. Sin embargo, si no es posible completar la interpretación de las solicitudes, o si las solicitudes involucran acciones que deben realizarse en el dispositivo móvil del usuario, START codifica las solicitudes del usuario en un idioma llamado Moebius, que ha sido diseñado para transmitir lenguaje natural solicitudes en diversas etapas de interpretación entre sistemas y dispositivos. Finalmente, el software que reside en el dispositivo móvil del usuario completa la

interpretación de las solicitudes de los usuarios, si es necesario, y realiza las acciones necesarias para cumplir con esas solicitudes.

Figura 2.6 Sistema StartMobile.

2.2.5 Finance ontology

El proyecto Finance Ontology [Mittal, 2018] amplia el sistema ILNBD de última generación y presenta una interfaz de diálogo a las bases de datos relacionales. La interfaz de diálogo permite a los usuarios explotar automáticamente el contexto semántico de la conversación mientras hacen consultas en lenguaje natural a través de SMBDR, lo que facilita la expresión de preguntas complejas de una manera natural y por partes. Propone nuevas técnicas basadas en la ontología para abordar cada uno de los desafíos específicos del diálogo, como la resolución de referencia conjunta, la resolución de puntos suspensivos y la desambiguación de consultas, usadas para determinar la intención general de la consulta del usuario. Demuestra la aplicabilidad y utilidad de la interfaz de diálogo en dos dominios diferentes (finanzas y sanidad).

Para superar esta deficiencia, se presenta una interfaz de diálogo para una base de datos que realiza automáticamente cada una de las tareas necesarias de comprensión contextual y resolución que explotan el conocimiento de la ontología y el SMBDR. Esto permite al usuario dividir intuitivamente una pregunta compleja en una serie de preguntas cortas y usar el contexto de las preguntas anteriores y sus respuestas para hacer preguntas de seguimiento de manera natural.

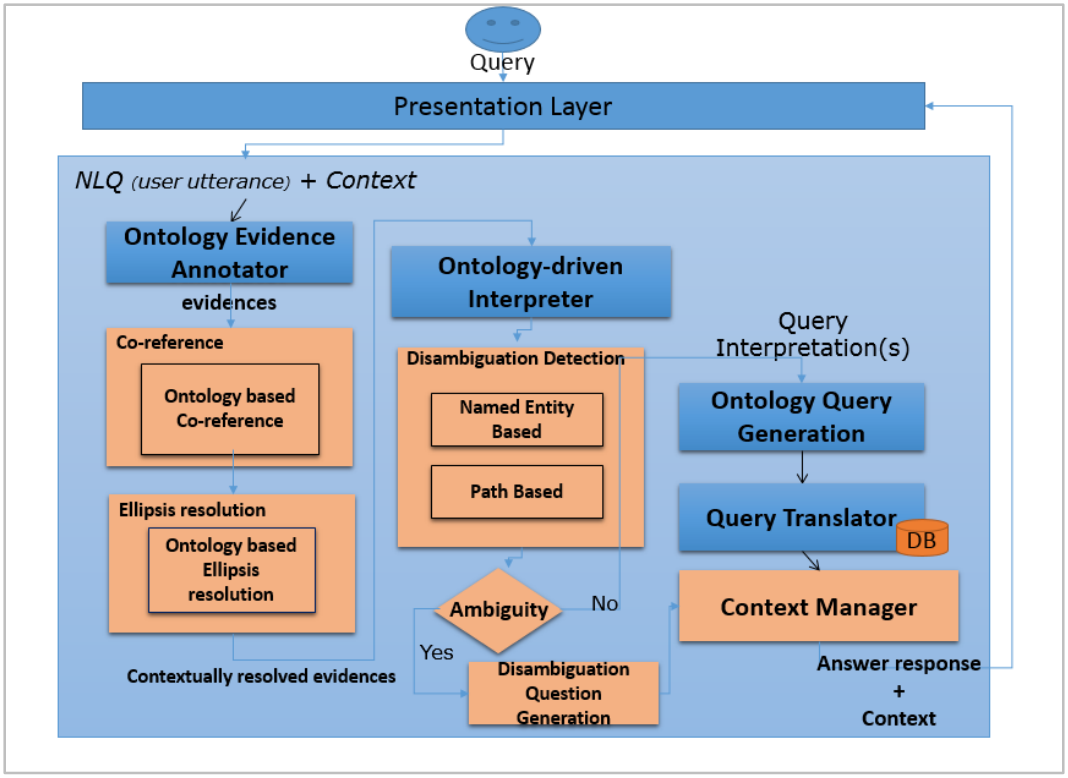


Figura 2.7 Arquitectura del sistema.

Descripción de los componentes del sistema:

Resolución de referencia conjunta.

Este componente es responsable de resolver las entidades de referencia conjunta, a saber, los pronombres. Mientras se resuelven tales correferencias, además de las características estándar de resolución de correferencia como criterios singulares / plurales, de género, criterios de vida / no vida, también se explota la coincidencia de nivel de elementos ontológicos que considera la igualdad en la jerarquía de herencia.

En caso de que haya un término de metadatos coincidente, que se refiere a múltiples valores en la respuesta anterior, estos valores se colocan en una lista que posteriormente crea un predicado "in" en la cláusula Where de la consulta SQL.

Resolución de puntos suspensivos.

Se requiere una resolución de puntos suspensivos para manejar consultas de seguimiento parcialmente especificadas (Expresiones sin oraciones). En tales casos, la consulta pasa por el procedimiento de anotación que identifica las porciones Select y Where de las cláusulas. La interfaz de diálogo identifica además la pregunta base del contexto que tiene la relación más cercana (calculada a través de la distancia del gráfico de ontología) y la anotación en la consulta.

Desambiguación.

Normalmente hay dos causas principales de ambigüedad en un diálogo: (i) ambigüedad entre entidades nombradas (por ejemplo, "Southwest" puede significar "Southwest Airlines" o "Southwest Securities"), o (ii) ambigüedad en la intención de la consulta (por ejemplo, "¿Qué son los préstamos de Citibank?", puede significar Citibank como prestamista o prestatario). Es esencial para un sistema de diálogo detectar estos casos de ambigüedad de manera adecuada, resolverlos automáticamente en la medida de lo posible utilizando el contexto de la sesión de diálogo y, como último recurso, sondear al usuario con una pregunta de aclaración.

Intención de resolución.

Este componente es responsable de mantener toda la información específica del contexto que se transmite a través de la capa de presentación. Es posible actualizar el estado de la conversación del usuario en forma de intentos que el usuario ha formulado en la pregunta anterior.

La interacción se inicializa con un contexto vacío. Cuando se detecta una pregunta del usuario, se crea un objeto de contexto con la pregunta y se empuja al contexto, representado como una pila. Si el sistema requiere una entrada adicional del usuario para la desambiguación, se crea un contexto de desambiguación con preguntas y opciones de desambiguación y se empuja al contexto. Se hace una pregunta de desambiguación en lenguaje natural y también se espera que la respuesta esté en el lenguaje natural. Además, el proceso de desambiguación puede no completarse mediante una sola pregunta y respuesta, sino que podría implicar una serie de preguntas de aclaración de seguimiento.

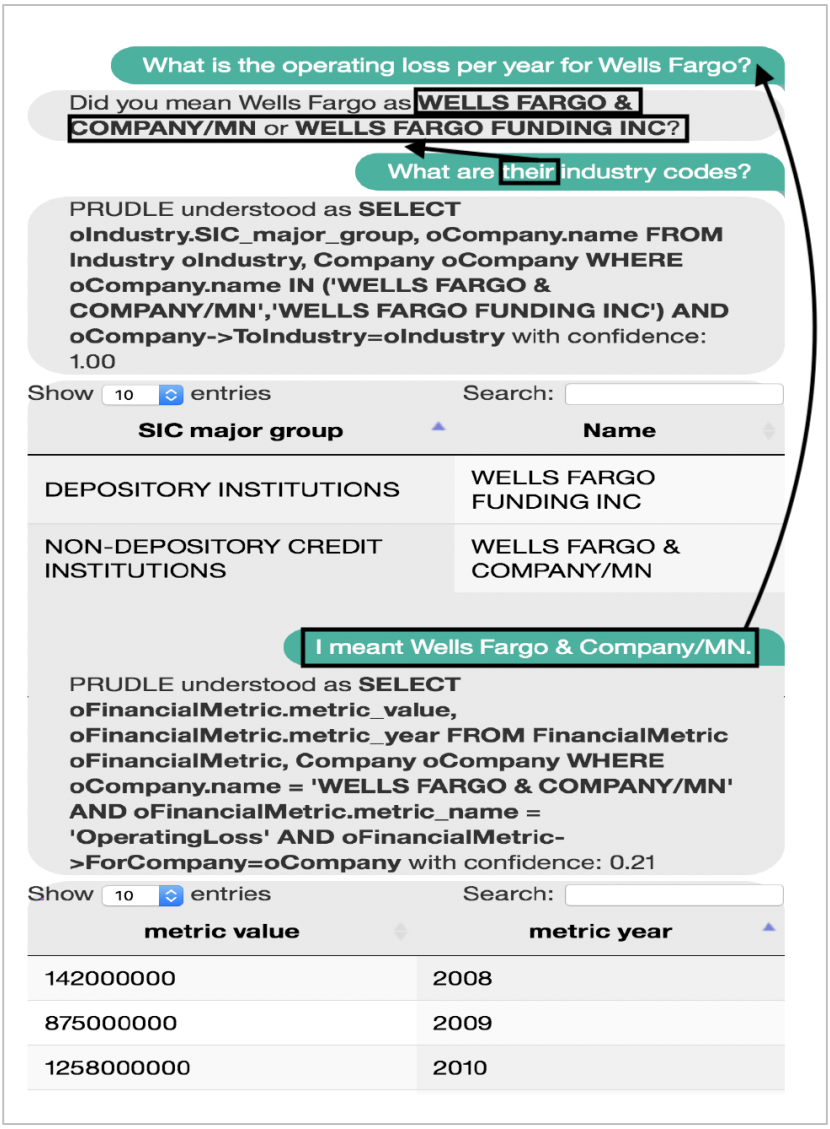


Figura 2.8 Interacción con la interfaz de diálogo resaltando ambigüedad.

2.2.6 Aneesah

El prototipo desarrollado Aneesah [Shabaz, 2015], es un sistema ILNBD con capacidades de conversación para proporcionar un entorno interactivo y amigable para ayudar a los usuarios con la información deseada almacenada en una base de datos relacional. Aneesah puede cumplir con los requisitos del usuario, guiar al usuario a su objetivo previsto y producir la información deseada de la base de datos relacional subyacente. La arquitectura propuesta se desarrolló mediante la adopción del enfoque de coincidencia de patrones.

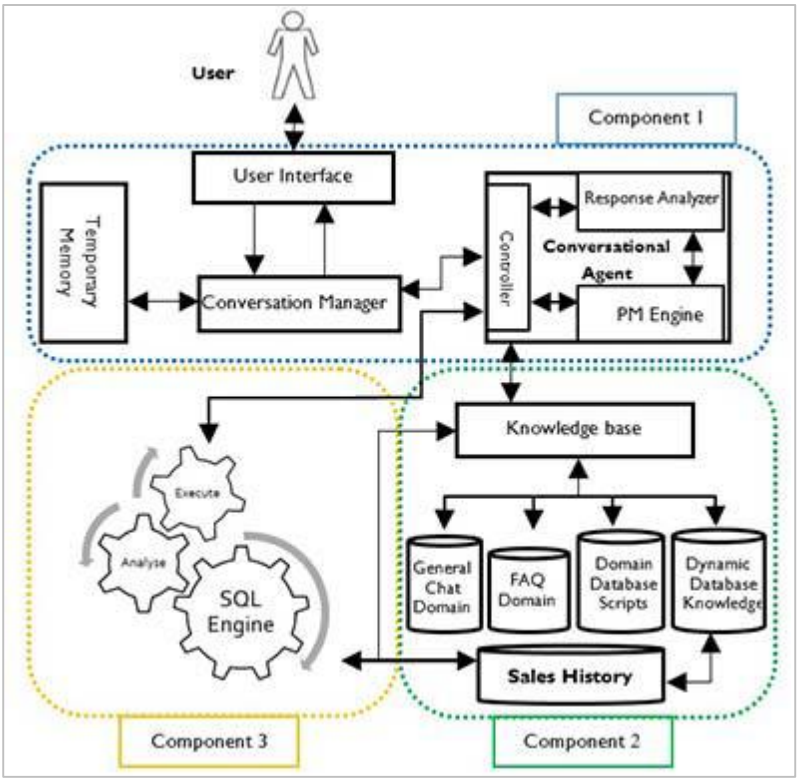


Figura 2.9 Arquitectura de la ILNBD Aneesah.

Aneesah emplea un nuevo marco construido con un lenguaje conversacional basado en lenguaje de script, base de conocimiento y motor SQL.

El **componente 1** de la arquitectura del sistema Aneesah se compone de los componentes Administrador de conversación, Interfaz de usuario, Memoria temporal y Agente de conversación (Controlador, Motor de coincidencia de patrones, Lenguaje de secuencias de comandos y Analizador de respuestas). El sistema Aneesah presenta un nuevo agente de conversación que es una parte fundamental de la arquitectura propuesta, que permite a los usuarios interactuar / conversar con el sistema Aneesah para fines de recuperación de información.

El **componente 2** de la arquitectura consiste en una base de conocimiento para servir como el cerebro de Aneesah que se ha desarrollado en base a las técnicas de ingeniería del conocimiento para trabajar específicamente con la base de datos del dominio (base de datos del historial de ventas). La base de conocimiento desarrollada se puede personalizar con ingeniería de conocimiento para trabajar con una base de datos diferente. Responsable de admitir una coincidencia de expresión de usuario basada en dos niveles en tres contextos de dominios diferentes. También es responsable de proporcionar conocimiento de dominio para llevar a cabo el mapeo de enunciados del usuario para que coincida con la información de la base de datos del dominio y el procesamiento de solicitudes de soporte con información mínima de los usuarios.

El **componente 3** de la arquitectura de Aneesah comprende el nuevo motor de consultas SQL con sus componentes. El motor de consultas SQL después de haber recibido la sintaxis / información relevante de la base de datos del componente controlador, es responsable de la formulación de consultas SQL en contraste con los enunciados de los usuarios. El motor SQL identifica el tipo y la naturaleza de las consultas que siguen mediante la activación del componente configurador de SQL.

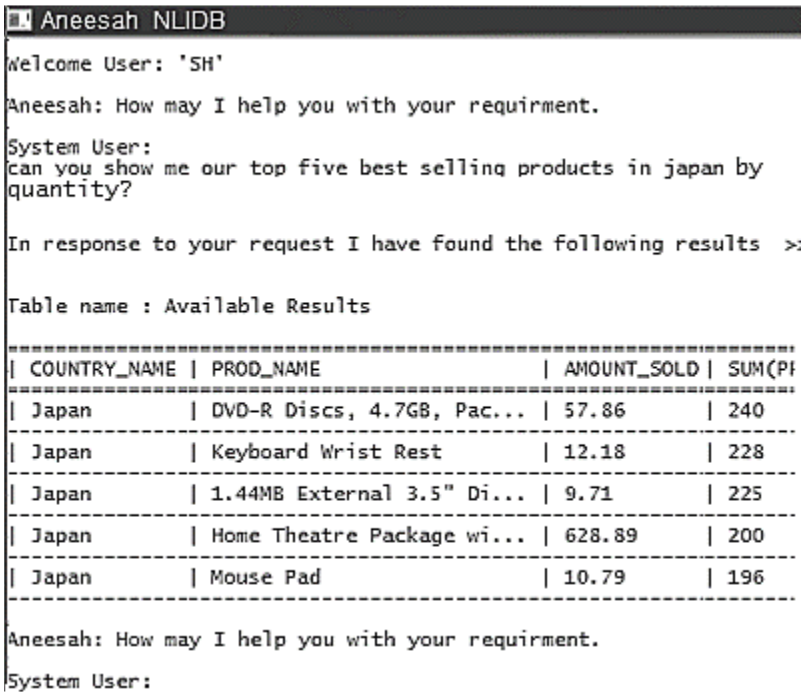


Figura 2.10 Interfaz de usuario Aneesah.

2.2.7 Conclusiones sobre trabajos relacionados

En la siguiente Tabla 2.1 se presenta un resumen de las interfaces con diálogo para aclaración de consultas.

Tabla 2.1 Tabla descriptiva de interfaces con diálogo para aclaración de consultas.

Nombre del Proyecto	Idioma	Tipo de Entrada	Tipo de Diálogo	Refinamiento de la consulta
QuestIO (2011)	Inglés	Escrita	Mediante lista de interpretaciones	Selecciona una opción de la lista de interpretaciones

FREyA (2011)	Inglés	Escrita	Mediante lista de interpretaciones	Selecciona una opción de la lista de interpretaciones
Aneesah(2015)	Inglés	Escrita	Agente de conversación	Se reinicia nuevamente la consulta
Análisis Semántico Basado en Reglas para ILNBD (2017)	Inglés	Escrita / Voz	No identificado	No identificado
StartMobile (2018)	Inglés	Escrita / Voz	Muestra un único resultado	Se reinicia nuevamente la consulta
Finance Ontology (2018)	Inglés	Escrita	Pregunta y da opciones de la interpretación	Se introduce una de las interpretaciones propuestas
Interfaz Propuesta	Español	Escrita	Mediante lista de interpretaciones	Da opción de cambiar o eliminar cada interpretación o agregar una nueva

Como se puede observar en la Tabla 2.1, todos estos proyectos con los que se compara esta propuesta están desarrollados para el idioma "inglés" y el tipo de entrada que reciben es "escrita", a excepción de "Análisis Semántico Basado en Reglas para ILNBD" y "StartMobile" que también pueden recibir la entrada por voz. Así también se puede observar que en la mayoría de los casos el tipo de diálogo entre la interfaz y el usuario se realiza mediante una lista de interpretaciones que propone la interfaz, con la finalidad de que el usuario seleccione únicamente una de estas opciones, y se muestre el resultado final de acuerdo a la interpretación de la interfaz; con la inconveniencia de que no se muestre el resultado deseado debido a la inexistencia de un refinamiento interactivo entre la interfaz y el usuario.

En lo que respecta a este proyecto, está desarrollado para el idioma español y el tipo de entrada que recibe es por escrito. En el diálogo se muestra en forma de lista la paráfrasis de la consulta SQL obtenida de la ILNBD, con la opción de refinar la consulta mediante un administrador de diálogo que le permite al usuario cambiar o eliminar cada interpretación, o incluso agregar una nueva.

Capítulo 3
Análisis y solución conceptual del problema

En este capítulo se describe el análisis de la ILNBD para web con la que se trabajó, y las arquitecturas de la ILNBD de la versión anterior y de la versión actual. Se describe la ampliación del diccionario de información semántica, así como la descripción del diseño conceptual de la paráfrasis y del administrador de diálogo como propósito de los objetivos específicos mencionados en la sección 1.1.

3.1 Análisis de la interfaz de lenguaje natural para bases de datos

Se realizó la revisión del núcleo de la interfaz de lenguaje natural para bases de datos versión web desarrollada en el ITCM, con el fin de comprender la funcionalidad de la misma. Cabe mencionar que de manera inicial esta interfaz fue implementada en código Java como primera versión de escritorio, posteriormente se complementó con código JavaScript, HTML y JSP para la versión Web. Por lo tanto, para hacer que la ILNBD funcione, es necesario hacer la instalación de un servidor "Apache Tomcat" y un Sistema Administrador de Base de Datos (SABD) "PostgreSQL" en la cual se debe restaurar las BDs utilizadas para esta versión.

Esta interfaz cuenta con los algoritmos para realizar procesos como lo son las conexiones a las BDs, el agregado de un usuario nuevo, el login de un usuario existente en el registro de la BD, los procesos de la traducción de una consulta en lenguaje natural a una consulta SQL, pasando por la fase del análisis léxico, sintáctico y semántico. Por otra parte también se realizó la identificación de las funciones de código que controlan los elementos gráficos, tales como ventanas, botones, listas desplegables, páginas, entre otros.

Otra de las finalidades de esta actividad fue identificar en que parte del código se integrarán los nuevos procesos para la aclaración de consultas de acuerdo a la estructura actual de la ILNBD.

3.2 Descripción de la arquitectura actual de la ILNBD

La arquitectura de la ILNBD versión web en la que se trabajó se muestra en la Figura 3.1, la cual cuenta con dos módulos principales:

- El primer módulo es la Interfaz de Configuración, diseñada para el control de usuarios, configuración automática (Generar dominio), afinación manual (Editor de dominio) y afinación con el wizard descritas en [González, 2018]:

  El submódulo de control de usuarios permite crear o iniciar sesión en la ILNBD y así poder utilizar todas las funciones que ofrece la interfaz de configuración.

  El submódulo de configuración automática (Generar dominio) permite generar un nuevo diccionario de información semántica de acuerdo con la base de datos que el usuario esté utilizando en ese momento.

  El submódulo de afinación manual (Editor de dominio) permite editar el diccionario de información semántica.

  El submódulo de consulta permite formular consultas en lenguaje natural, además tiene la opción de utilizar la afinación con el wizard.

- El segundo módulo es la Interfaz de Consulta, diseñada para realizar consultas en lenguaje natural, devolviendo una consulta en SQL (inglés) y los resultados en forma tabular de dicha consulta.

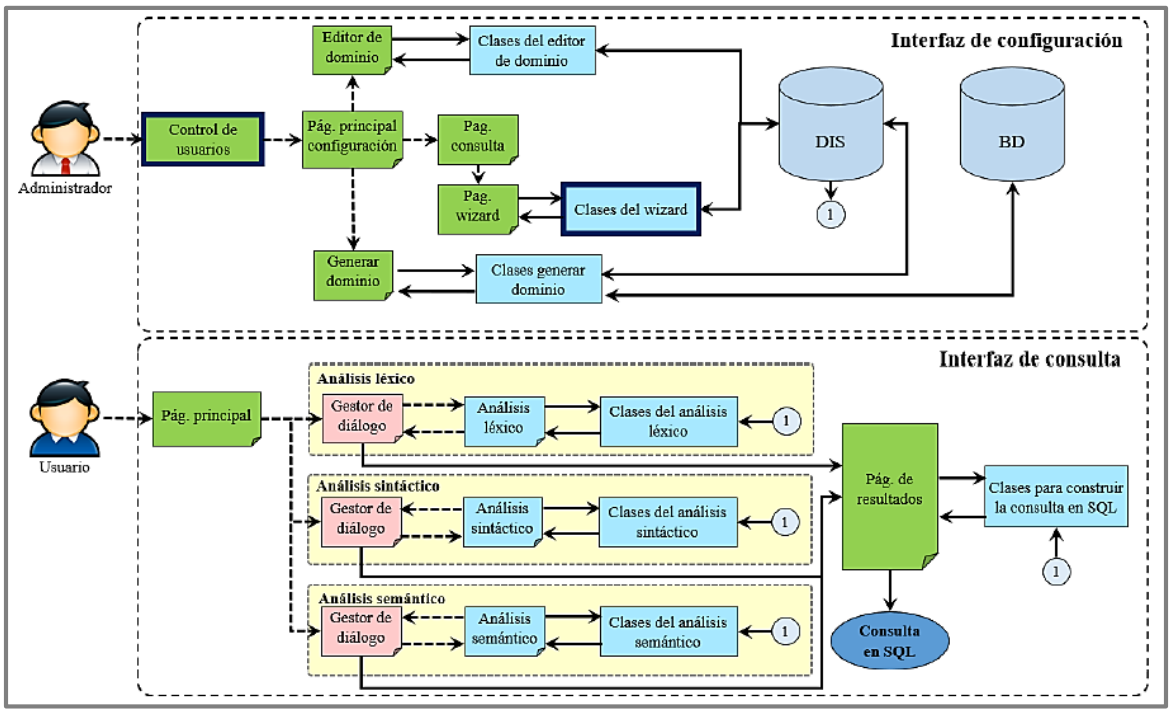


Figura 3.1 Arquitectura de la ILNBD versión para web [González, 2018].

3.3 Descripción de la nueva arquitectura de la ILNBD

La implementación del módulo de "Aclaración de Consulta" se integró en el módulo de Interfaz de Consultas de Lenguaje Natural, como se muestra en la arquitectura de la Figura 3.2. Este módulo consiste primeramente en realizar una paráfrasis de la consulta SQL que se obtiene en el proceso de traducción de la consulta en lenguaje natural que ingresa el usuario, para el cual se presentan las siguientes dos opciones:

- Un botón para continuar con el proceso, el cual muestra los resultados finales obtenidos de la base de datos consultada.
- Otro botón para cambiar la interpretación de la interfaz, en el cual por medio de un administrador de diálogo le permitirá al usuario agregar, editar o eliminar cada interpretación para aclarar la consulta.

Sin embargo, para lograr la paráfrasis de la consulta SQL se consideró también la ampliación del diccionario de información semántica (DIS).

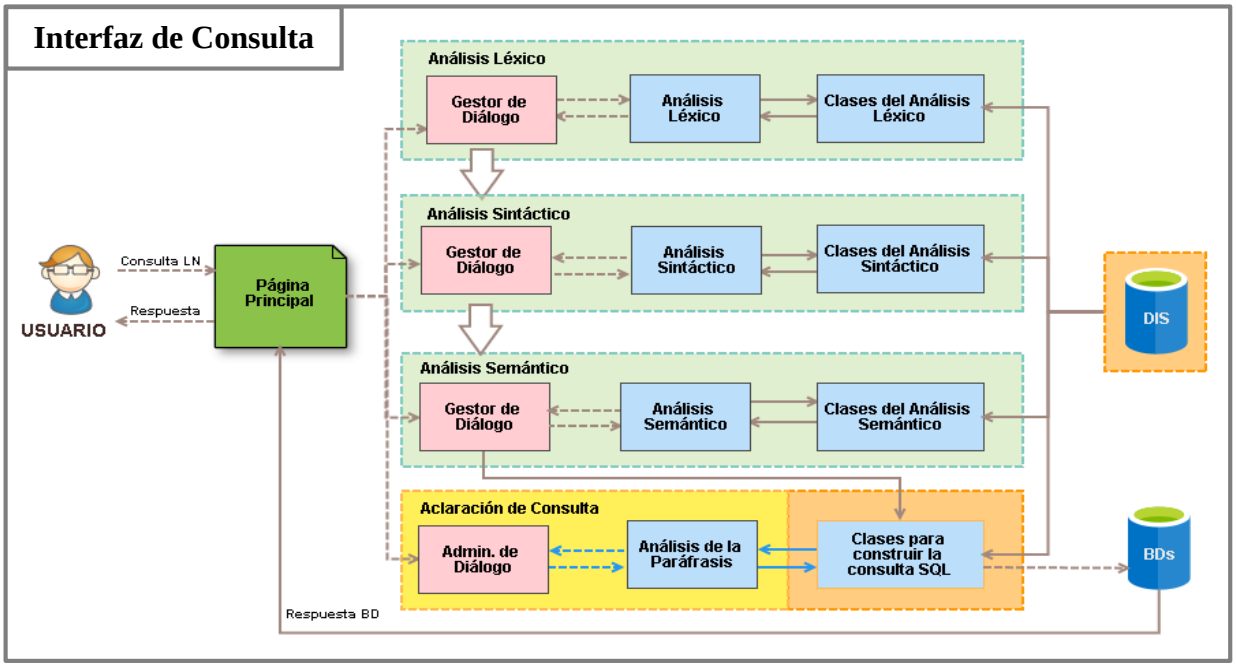


Figura 3.2 Nueva arquitectura de la interfaz de consulta.

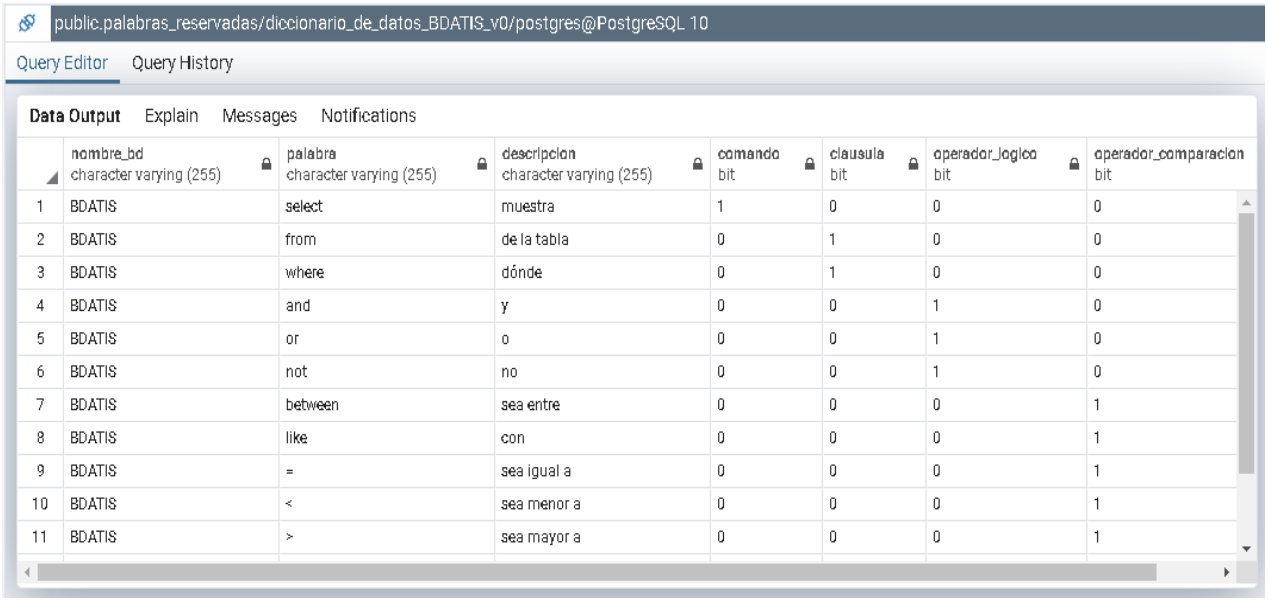
3.4 Ampliación del diccionario de información semántica (DIS)

El lenguaje SQL está compuesto de palabras reservadas, es decir, palabras especiales que ejecutan operaciones SQL. Por lo cual, para el proceso de implementación de la paráfrasis de la consulta SQL se consideró la ampliación del DIS, esto consistió en agregar

una nueva tabla con las palabras reservadas necesarias para la interpretación, con su respectiva descripción y el componente del lenguaje SQL al que pertenecen, para así lograr la interpretación de los elementos solicitados en el comando SELECT y las condiciones identificadas en la cláusula WHERE de la consulta SQL, de tal modo que el usuario pueda interactuar y mantener un proceso diálogo para aclarar la consulta.

En la Tabla 3.1 se muestra la estructura y contenido de la tabla "*palabras_reservadas"* agregada en el DIS. Esta ampliación es realizada con la finalidad de identificar cada palabra de la consulta SQL, si pertenece a un tipo de cláusula u operador. Si la palabra corresponde a un comando o una cláusula se podrá identificar los datos que la componen para la selección y manipulación de los mismos (elementos y condiciones), si corresponde a un operador se podrá obtener su descripción para complementar la interpretación.

Tabla 3.1 Estructura de la tabla *palabras_reservadas*.


public.palabras_reservadas/diccionario_de_datos_BDATIS_v0/postgres@PostgreSQL 10

Query Editor | Query History

Data Output | Explain | Messages | Notifications

	nombre_bd character varying (255)	palabra character varying (255)	descripcion character varying (255)	comando bit	clausula bit	operador_logico bit	operador_comparacion bit
1	BDATIS	select	muestra	1	0	0	0
2	BDATIS	from	de la tabla	0	1	0	0
3	BDATIS	where	dónde	0	1	0	0
4	BDATIS	and	y	0	0	1	0
5	BDATIS	or	o	0	0	1	0
6	BDATIS	not	no	0	0	1	0
7	BDATIS	between	sea entre	0	0	0	1
8	BDATIS	like	con	0	0	0	1
9	BDATIS	=	sea igual a	0	0	0	1
10	BDATIS	<	sea menor a	0	0	0	1
11	BDATIS	>	sea mayor a	0	0	0	1

Contenido y descripción de las columnas de la nueva tabla:

Nombre: El valor de esta columna contiene el nombre de la base de datos.

Palabra: El valor de esta columna corresponde a una palabra reservada que forma parte de la sintaxis de una consulta SQL.

Descripción: Descripción del comando, cláusula u operador (en idioma español).

Comando: Permiten generar consultas para ordenar, filtrar y extraer datos de la BD.

Cláusula: Son las condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Operador (Lógico y Comparación): Son combinaciones de caracteres que se utilizan para realizar asignaciones o comparaciones entre datos.

3.5 Diseño conceptual de la paráfrasis

Para la funcionalidad del módulo de aclaración de consultas, se propuso que después de los procesos encargados de realizar la traducción del LN, el cual da como resultado una consulta SQL, se mostrara una paráfrasis de dicha consulta antes de ser enviada a la base de datos. Este proceso consiste en identificar los elementos pertenecientes al comando SELECT e identificar las condiciones de la cláusula WHERE, para que de este modo se pueda desplegar en forma de una lista un apartado de *Muestra* de los elementos a presentar y otra lista para el apartado *Tal que* correspondiente a las condiciones a cumplir.

A continuación se muestra un ejemplo del diseño conceptual de la paráfrasis:

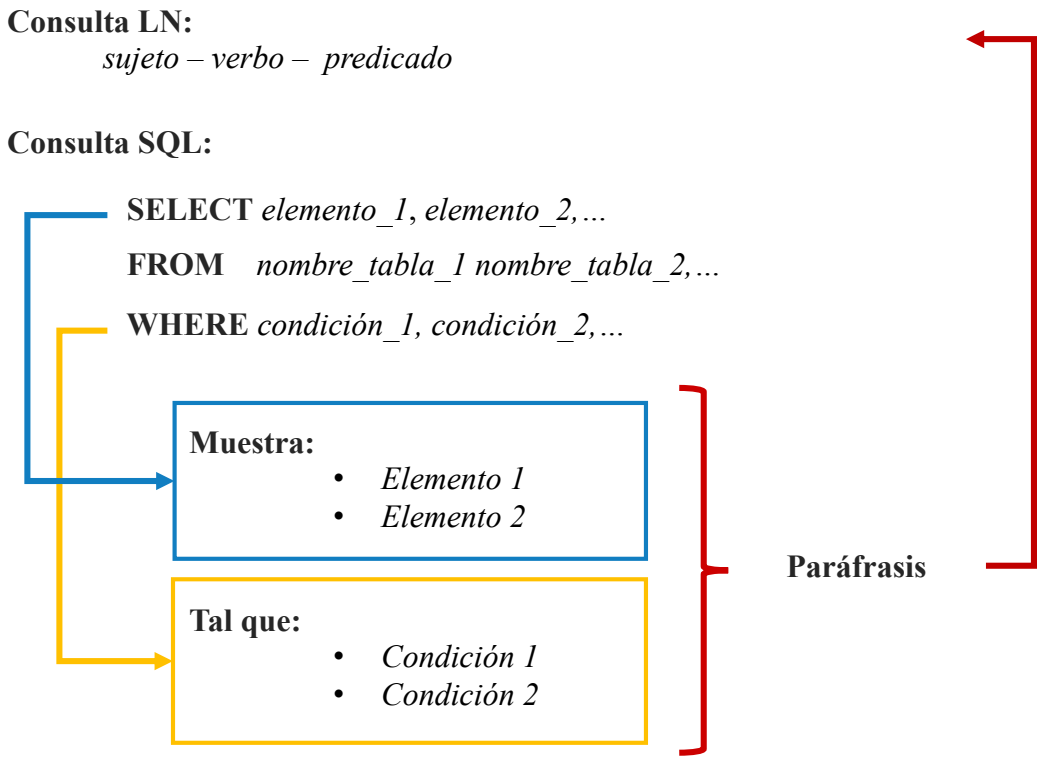


Figura 3.3 Diseño conceptual de la paráfrasis.

El objetivo principal de realizar la paráfrasis es que el usuario pueda visualizar si su consulta fue interpretada de manera correcta por la interfaz para así poder continuar con el proceso y mostrar el resultado de la consulta solicitada, o en caso contrario, cambiar la

interpretación por falta de información proporcionada o por algún dato ambiguo ingresado por el mismo usuario.

Por ejemplo, en la consulta *Muéstrame la tarifa de viaje del vuelo 137 que sale por la mañana* no se especifica el tipo de vuelo, sencillo o redondo, así como tampoco se proporciona un horario específico de la salida del vuelo. Por tanto, de acuerdo a los procesos de traducción del LN y a la consulta SQL obtenida por la interfaz, se realiza la paráfrasis (Figura 3.4.) de la misma como se mencionó anteriormente, en este caso se muestra todas las diferentes tarifas de vuelo encontradas en la BD, con la restricción de que sea del vuelo número 137 y con la especificación del horario de la *mañana* interpretada por la interfaz.

Consulta LN:

Muéstrame la tarifa de viaje del vuelo 137 que sale por la mañana.

Traducción SQL obtenida de la interfaz:

SELECT fare.rnd_trip_cost, 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 departure_time BETWEEN 0000 AND 1159 AND flight.flight_number = 137;

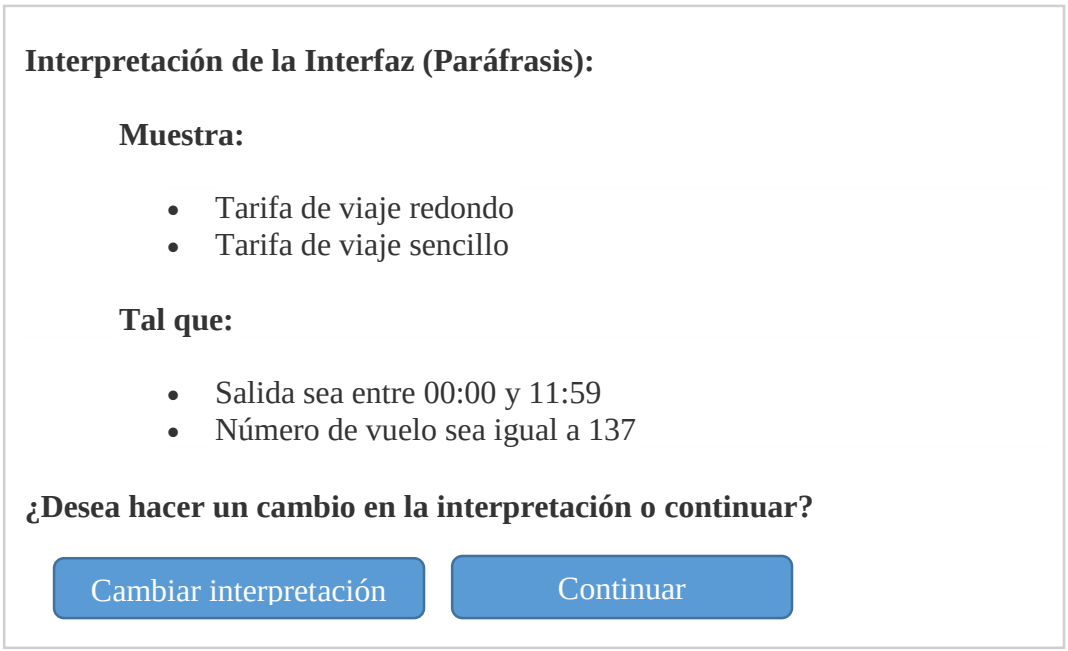


Figura 3.4 Diseño de la consulta SQL parafraseada.

3.6 Diseño conceptual del administrador de diálogo

Con el propósito de que el usuario final pueda aclarar la interpretación de la interfaz, se da la opción de que mediante el botón "Cambiar la interpretación" se despliegue la ventana *administrador de diálogo* como se muestra en la Figura 3.5. Con este administrador el usuario puede cambiar, eliminar o agregar uno o más elementos y/o condiciones de la paráfrasis según considere el usuario necesario para aclarar la consulta.

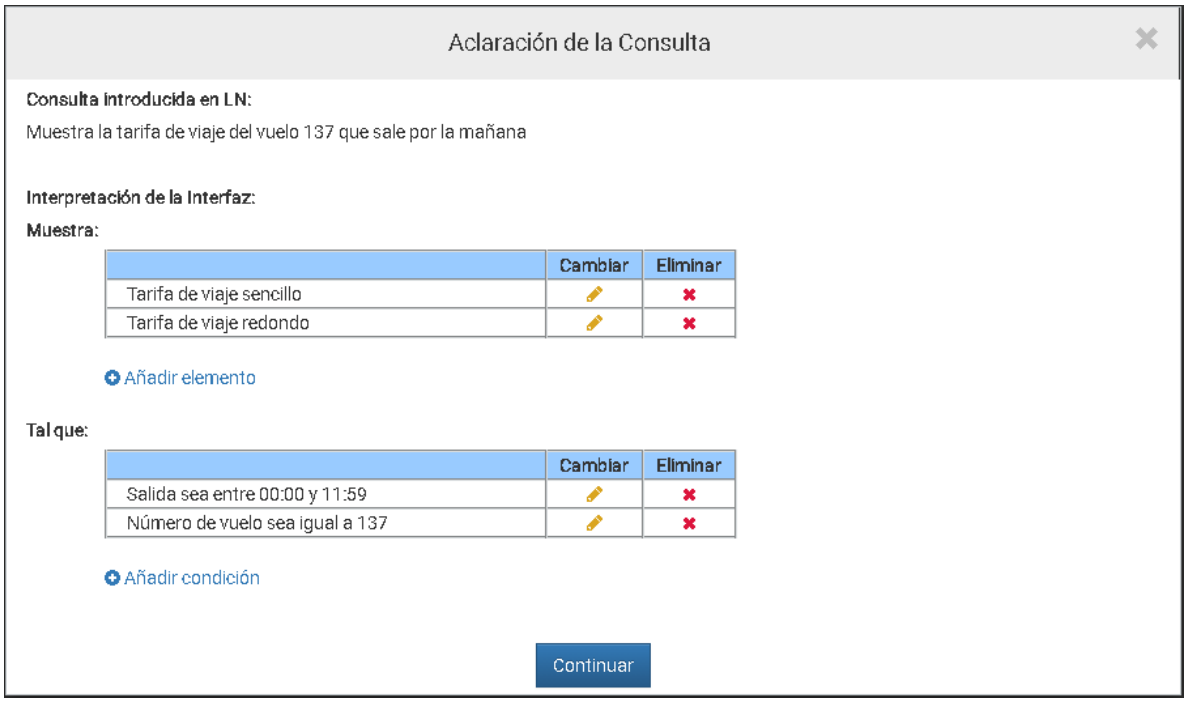


Figura 3.5 Diseño del administrador de diálogo.

El diseño de la ventana de la Figura 3.5 muestra la consulta original que ingresa el usuario, así como también, la interpretación de la interfaz (paráfrasis) de la consulta; que consta de dos apartados (*Muestra* y *Tal que*), la cual se detalla en la Sección 3.5 *Diseño conceptual de la paráfrasis.* Cada apartado es presentado por medio de una tabla que divide los elementos y las condiciones de cada uno, con tres posibles opciones:

- **Cambiar:** Representado por un icono en forma de lápiz, esta opción habilita un recuadro en la misma celda para poder editar la interpretación de manera escrita.

- **Eliminar:** Representado por un icono en forma de 'x' que tiene como función borrar la fila seleccionada.

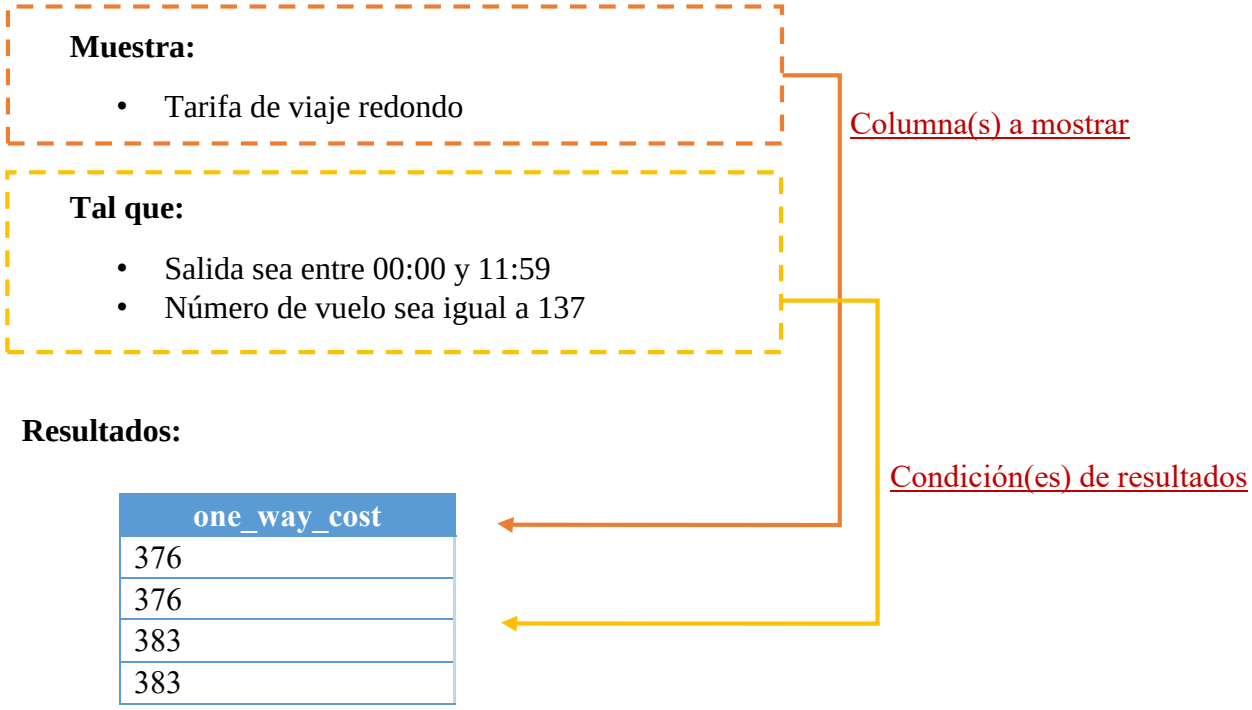
- **Añadir:** Al seleccionar esta opción se incorpora una nueva fila en la tabla del apartado correspondiente para agregar una nueva aclaración.

En la parte final de la ventana se muestra un botón de *Continuar*, que por medio del evento *click()*, envía la información de los elementos y condiciones de cada apartado a un proceso de validación, si éstos no se pueden interpretar por la interfaz, se notificará al usuario por medio de diferentes diálogos de aclaración hasta que éstos sean válidos para generar una consulta SQL, ésta será parafraseada y se mostrará al usuario junto con el resultado final de la consulta aclarada como se muestra en la Figura 2.

Consulta LN:

Muéstrame la tarifa de viaje del vuelo 137 que sale por la mañana.

Interpretación de la Interfaz:



Resultados:

one_way_cost
376
376
383
383

Figura 3.6 Esquema de la interpretación final y resultados.

Capítulo 4
Desarrollo del módulo de aclaración

En este capítulo se presenta la parte importante del desarrollo del módulo de aclaración de la ILNBD para web. Se describe la parte de la implementación de la paráfrasis de la consulta SQL obtenida del traductor de la interfaz, así como la implementación del administrador de diálogo que consiste en un proceso de edición, eliminación y agregación de la paráfrasis, mediante una interacción entre usuario-maquina.

4.1 Implementación de la paráfrasis

En la implementación de la paráfrasis se diseñó la interfaz en código HTML, como se presenta en la Figura 4.1, se muestra la consulta ingresada por el usuario en LN (idioma español), la paráfrasis de la consulta SQL en forma de dos listas (*Muestra* y *Tal que)*, también se muestra una pregunta de confirmación para el usuario que incluye dos botones; uno es para "Continuar" con el proceso y obtener un resultado de manera directa si el usuario así lo desea, y otro para "Cambiar interpretación" mediante un administrador de diálogo.

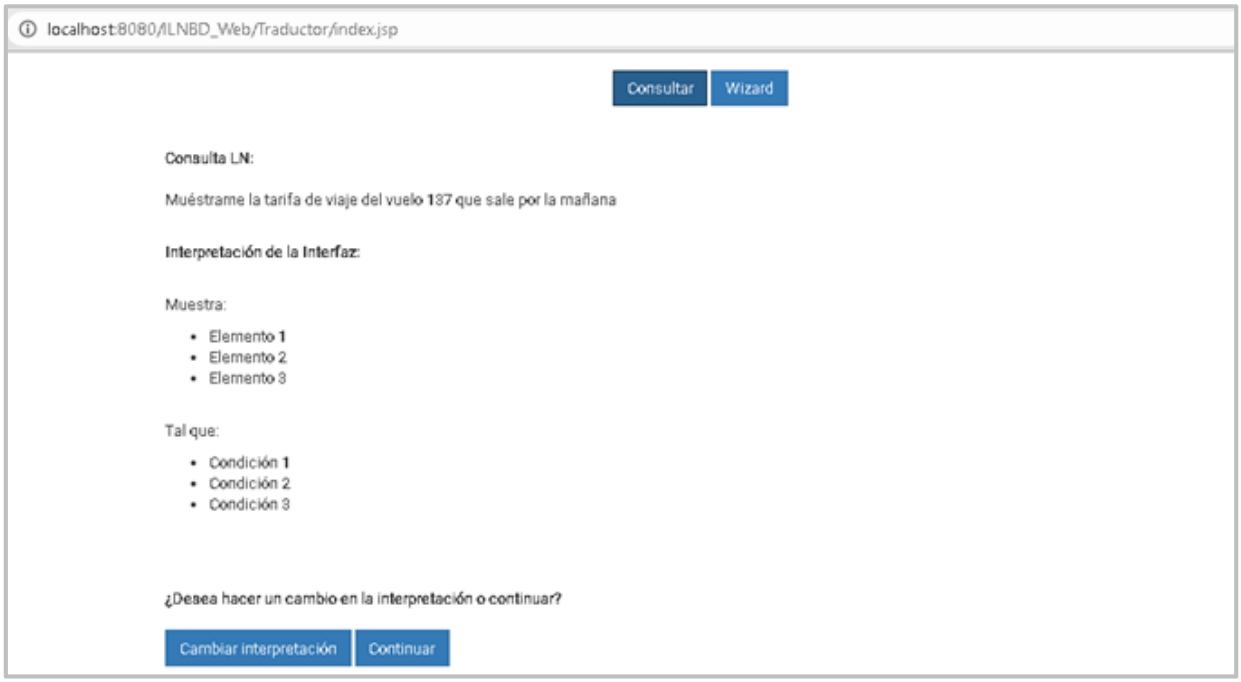


Figura 4.1 Diseño de la interpretación de la interfaz.

Una vez que el usuario ha ingresado la consulta en LN (idioma español), ésta es procesada por el traductor de la interfaz que da como resultado una consulta SQL. De acuerdo a la propuesta de la nueva arquitectura de la interfaz de consulta (Figura 3.2), posteriormente esta consulta SQL obtenida es enviada al procedimiento del análisis de la paráfrasis.

El procedimiento del análisis de la paráfrasis inicialmente consiste en etiquetar cada palabra o carácter (omitiendo las referencias de tablas, comas y el punto y coma) para identificar si corresponde a una tabla, columna, valor de búsqueda (?) o algún tipo de componente SQL (comando, cláusula, operador lógico, operador de comparación), de acuerdo a la base de datos del DIS, como se muestra en la Algoritmo 4.1.

Algoritmo 4.1 Pseudocódigo del etiquetado de la consulta SQL

```
1   Q // Consulta de entrada SQL
2   n // Número total de tokens
3   L // Lista de etiquetas
4   for i = 0, …, n-1 do
5       if esComponenteSQL(Qi)
6           L ← etiquetaComponente(Qi)        // Etiquetar como un Componente SQL
7       else
8           if esColumna (Qi)
9               L ← etiquetaColumna (Qi)      // Etiquetar como Columna
10          else
11              if esTabla (Qi)
12                  L ← etiquetaTabla (Qi)    // Etiquetar como Tabla
13              else
14                  L ← etiquetaValorBusqueda (Qi)    // Etiquetar como Valor de búsqueda
15              endif
16          endif
17      endif
18  endfor
```

Posteriormente se hace un recorrido de la lista obtenida de las etiquetas para extraer el rango de posiciones de donde se pueden identificar los elementos correspondientes al comando SELECT y las condiciones correspondientes a la cláusula WHERE, mostrado en el Algoritmo 4.2.

Finalmente, se ejecuta un proceso para identificar los componentes y construir la paráfrasis de cada apartado (*Muestra* y *Tal que*) que se muestra al usuario como la interpretación de la interfaz.

Algoritmo 4.2 Pseudocódigo de la posición de columnas y condiciones

```
1   Q // Consulta de entrada SQL
2   j // Posición inicial Select
3   k // Posición final Select
4   p // Posición inicial Where
5   q // Posición final Where
6   L // Lista de etiquetas
7   for i = 0, …, size(L) - 1  do
8       if esComponente(Li) and esTokenSelect(Qi)
9           j ← i       // Guardar posición inicial Select
10      else
11          if esClausula(Li) and esTokenFrom(Qi)
12              k ← i       // Guardar posición final Select
13          else
14              if esClausula(Li) and esTokenWhere(Qi)
15                  p ← i     // Guardar posición inicial Where
16                  q ← size(L) - 1      // Guardar posición final Where
17          endif
18      endif
19  endfor
```

4.1.1 Paráfrasis del comando Select

En la implementación de la paráfrasis del comando SELECT, la cual corresponde al apartado *Muestra* en la interfaz de usuario, posterior al etiquetado de la consulta y con base al rango de posiciones, se identifican todos los tokens con la etiqueta *columna* que se encuentran entre las palabras reservadas SELECT y FROM, sin considerar las que posiblemente también existan en la cláusula WHERE, como se muestra en un estructura general en el Algoritmo 4.3, en la cual también se manda a llamar la función *InterpretacionColumna(),* ésta devuelve la paráfrasis de la columna con ayuda del diccionario de información semántica. Por otra parte, en esta función también se considera el caso de la consulta de todas las columnas de una tabla (SELECT *).

Cabe mencionar que los tokens con la etiqueta *columna* se verifican que correspondan a una columna de las tablas existentes en la BD a consultar durante el procedimiento del etiquetado.

Algoritmo 4.3 Pseudocódigo de la identificación de columnas

```
1   L // Lista de etiquetas
2   C // Lista de interpretación columna
3   S // Lista total de interpretaciones Select
4   for i = 0, …, n-1 do
5       if i > j and i < k      //Verifica rango de posiciones
6           C ← InterpretacionColumna(Qi)  // Interpretar columna
7           for i = 0, …, size(C) - 1 do
8               S ← C       // Guardar interpretación de la columna
9           endfor
10      endif
11  endfor
```

En la Figura 4.2 se muestra la interpretación de la interfaz del comando SELECT, en este ejemplo se identifican dos elementos columna a mostrar *rnd_trip_cost* y *one_way_cost*, a los cuales durante el proceso se les omite la referencia de la tabla (*fare.*), la paráfrasis de cada elemento de la consulta SQL se pueden apreciar marcados con diferente tonalidad.

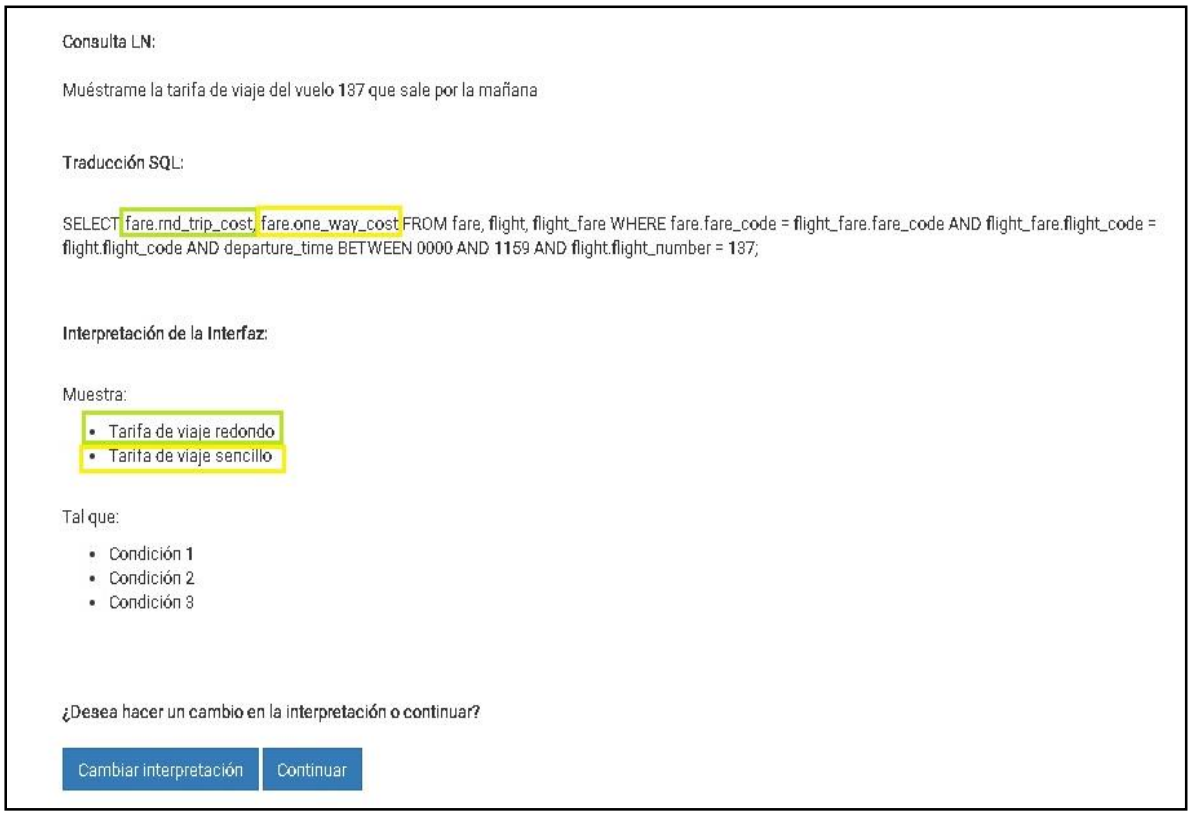


Figura 4.2 Paráfrasis del comando SELECT.

4.1.2 Paráfrasis de la cláusula Where

En la implementación de la paráfrasis de la cláusula WHERE, la cual corresponde al apartado *Tal que* en la interfaz de usuario, posterior al etiquetado de la consulta y con base al rango de posiciones, se identifican las condiciones y se evalúan con base a reglas que cumplen con la sintaxis de una condición (Algoritmo 4.4), de acuerdo a la estructura general de consultas en SQL de la ILNBD.

Regla de condición que no se considera mostrar al usuario:

- *columna + operador de comparación + columna*

En esta regla se hace caso omiso de mostrar la igualdad entre columnas identificadoras, ya que estas se consideran innecesarias para mostrar al usuario debido a que es una relación de tablas las cuales el usuario desconoce.

Reglas de condición que si se consideran mostrar al usuario:

- *columna + operador_comparación + valor_búsqueda_1 (?) + operador_lógico + valor_búsqueda_2 (?)*

- *columna + operador_comparación + valor_búsqueda (?)*

Algoritmo 4.4 Pseudocódigo de la identificación de condiciones

```
1   W // Lista total de interpretaciones Where
2   for i = 0, …, n-1 do
3       if p>0 and i > p and i <= q
4           if esColumna(Li) and esOperadorIgual(Li+1) and esColumna (Li+2)
5               i ← i+2   // Incrementa valor de i
6           else
7               if esColumna(Li) and esOperadorSQL(Li+1) and esValorBusqueda (Li+2)
8                   W ← InterpretacionCondicion(Qi)  // Guardar interpretación de la condición
9                   i ← i+2    // Incrementa valor de i
10              else
11                  if esColumna(Li) and esOperadorBetween(Li+1) and esValorBusqueda1(Li+2)
12                    and esValorBusqueda2(Li+4)
13                     W ← InterpretacionCondicion(Qi)  // Guardar interpretación de la condición
14                     i ← i+4    // Incrementa valor de i
15                  endif
16              endif
17          endif
18      endif
19  endfor
```

En la Figura 4.3 se muestra la interpretación de la interfaz de la cláusula WHERE, en este ejemplo se identifican dos condiciones que se consideran mostrar, omitiendo la igualdad de columnas identificadoras, la paráfrasis de cada condición de la consulta SQL se pueden apreciar marcados con diferente tonalidad.

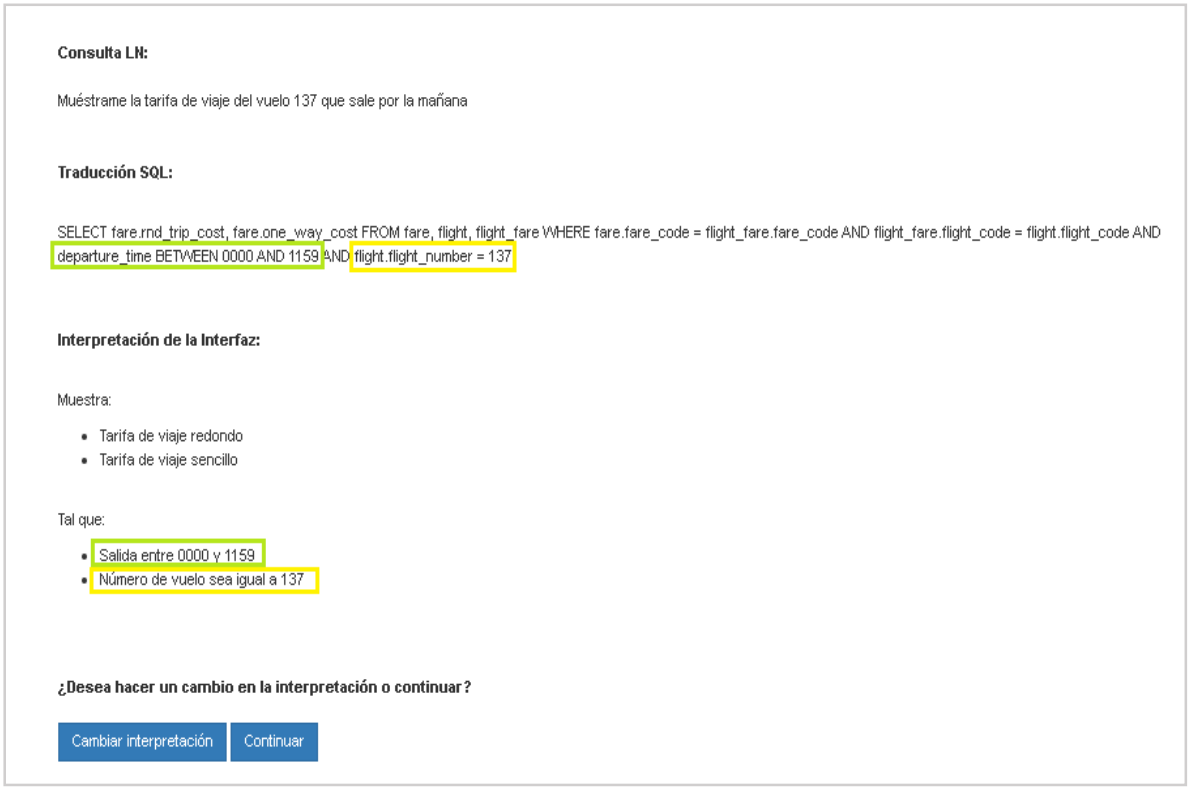


Figura 4.3 Paráfrasis de la cláusula WHERE.

Para la implementación del análisis de la paráfrasis de la consulta SQL se agregó la clase *Parafrasis.java.* A continuación se describen algunos de los métodos que integran esta clase:

- **ArrayList getElementos():** Envía una lista de los elementos como resultado final obtenido.

- **ArrayList getCondiciones():**Envía una lista de las condiciones como resultado final obtenido.

- **Parafrasis(Connection conLexicon, Connection conVerbos, Connection conDD, Connection DB):** Recibe las conexiones a las BDs del lexicón, verbos y del diccionario de información semántica.

- **Consulta procesar(Consulta consultaSQL):** Recibe la consulta SQL obtenida de la interfaz, para posteriormente procesarla a otros métodos de la misma clase.

- **ArrayList <String> getEtiquetas(Consulta consultaModificada):** Hace un recorrido de la consulta SQL para identificar y etiquetar cada token como comando, clausula, operador lógico, operador de comparación, tabla, columna o como valor de búsqueda (?).

- **getPosicion(Consulta consultaModificada, ArrayList listEtiqueta):** Hace un recorrido de la consulta para guardar las posiciones de donde se obtendrán los elementos y condiciones.

- **String verificarTablaColumn(String token):** Verifica si el token recibido pertenece a una tabla o columna.

- **Consulta identificacionColumnasyCondiciones(Consulta consultaModificada, ArrayList listEtiqueta):** De acuerdo a las posiciones obtenidas de *getPosicion* se validan los tokens correspondientes al Select y Where de la consulta SQL, para posteriormente realizar y guardar la interpretación.

- **String cleanToken(String cadena):** Depura caracteres y la referencia de tabla detectados en los tokens de la consulta.

- **String InterpretacionColumna(String token):** Interpreta cada token recibido como elemento.

- **String InterpretacionCondicion(String token):** Interpreta cada token recibido como condición.

- **imprimeListaColumn():** Imprime la lista final obtenida de los elementos.

- **imprimeListaCondiciones():** Imprime la lista final obtenida de las condiciones.

4.2 Implementación del administrador de diálogo

En esta sección se describen los métodos implementados para el administrador de diálogo presentado en la sección *3.6 Diseño conceptual del administrador de diálogo*, así también se detallan las verificaciones consideradas que se realizan en el apartado de *Muestra* y *Tal que* para la aclaración de la consulta.

4.2.1 Clase del administrador de diálogo

Para la implementación del administrador de diálogo se agregó la clase *AdminDialogo.java.* A continuación se describen algunos de los métodos que integran esta clase:

- **AdminDialogo (String BD, String DIS):** Recibe la conexión a la BD de consulta y del diccionario de información semántica.

- **JSONObject validaDatos(String[] data_element, String[] data_condition):** Recibe la lista de los elementos y condiciones finales que se envían uno a uno a diferentes métodos para ser verificados, si éstos son válidos y no hay ningún mensaje de aclaración por la interfaz se construye una nueva consulta en SQL.

- **validaColumna (String palabra):** Recibe la(s) palabra(s) de la lista de elementos, verifica que existan con una columna de la BD a consultar, tal como se ingresó, en caso contrario, se descompone por tokens y se verifica la relación con alguna columna.

- **validaCondicion (String cadena):** Recibe una cadena de la lista de condiciones y ésta es descompuesta por tokens para ser etiquetados como columna, tabla, palabra reservada (comando, clausula, operador lógico, operador de comparación), palabra inútil o valor de búsqueda. Posteriormente todas las palabras que fueron marcadas con la misma etiqueta de *columna* y *operador comparación* son enviadas a un método para seleccionar una columna y un operador de comparación. Finalmente verifica que la condición cumpla con la estructura.

- **isColumn(String token, String befor_label):** Verifica si el token recibido es o no una columna.

- **isPalabraReservada(String token, String op):** Verifica si el token recibido es o no una palabra reservada.

- **isPalabraInutil(String token):** Verifica si el token recibido es o no una palabra inútil.

- **isTabla(String token):** Verifica si el token recibido es o no una tabla.

- **getColumna(String columns):** Recibe todos los tokens etiquetados como *columna* para obtener una sola columna como resultado final.

- **getOperadorComparacion(String opsComp):** Recibe todos los tokens etiquetados como *operador comparación* para obtener un solo operador de comparación como resultado final.

- **getRelaciones(ArrayList<String> tablas):** Identifica la relación entre tablas almacenadas con relación a las columnas obtenidas.

- **procesarParafrasis(String consultaGenerada):** Recibe la consulta SQL generada en el método *validaDatos()* para ser enviada al proceso de *Parafrasis()*.

4.2.2 Validación del apartado *Muestra*

En el proceso de validación de elementos del apartado *Muestra*, se verifica al inicio que se haya ingresado al menos un elemento a mostrar, en caso de que no se encuentre ninguno, esto se notifica al usuario con un diálogo de aclaración como se muestra en la Figura 4.4.

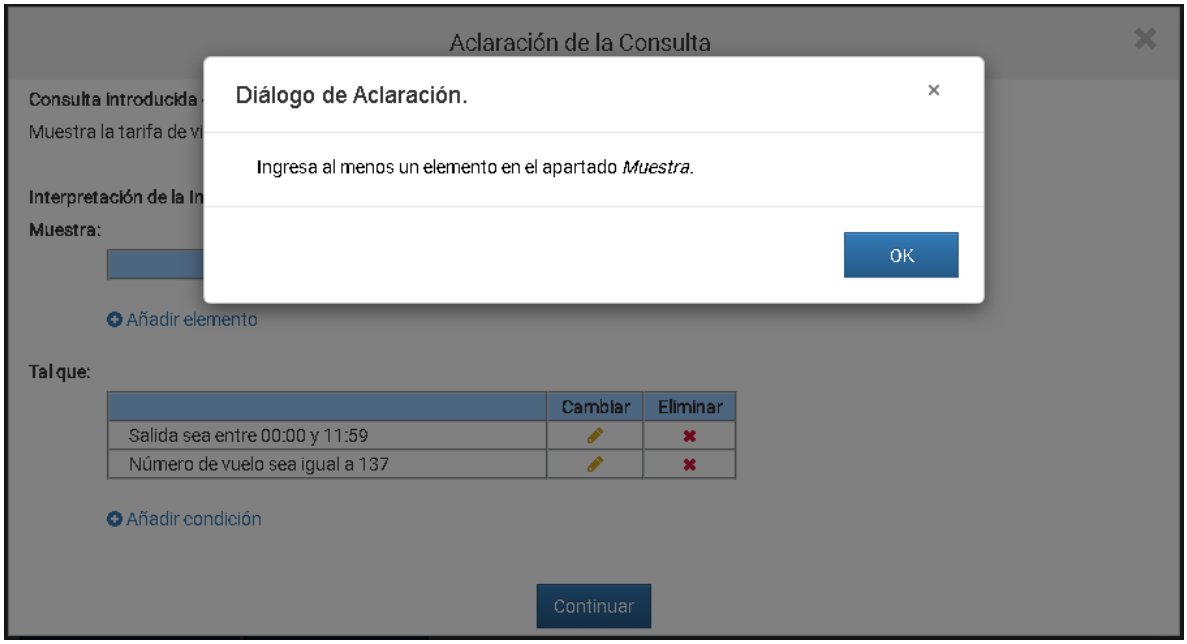


Figura 4.4 Diálogo de aclaración: Debe ingresar al menos un elemento.

Cuando se ingresan uno o más elementos, se verifican que existan éstos como una columna en la BD que se consulta, uno a uno, tal como se ingresó en el apartado de *Muestra*, en caso contrario, se descompone por palabra y se verifica la relación con alguna columna. Si el elemento se encuentra asociado con una columna, el proceso continúa, por el contrario, si se encuentra más de una asociación con otras columnas se muestra un diálogo de aclaración con el mensaje que se muestra en la Figura 4.5, donde se indica el elemento a aclarar, con la posible opción de visualizar un listado de sugerencias con relación al elemento ingresado, esto con la finalidad de que el usuario pueda precisar el elemento solicitado.

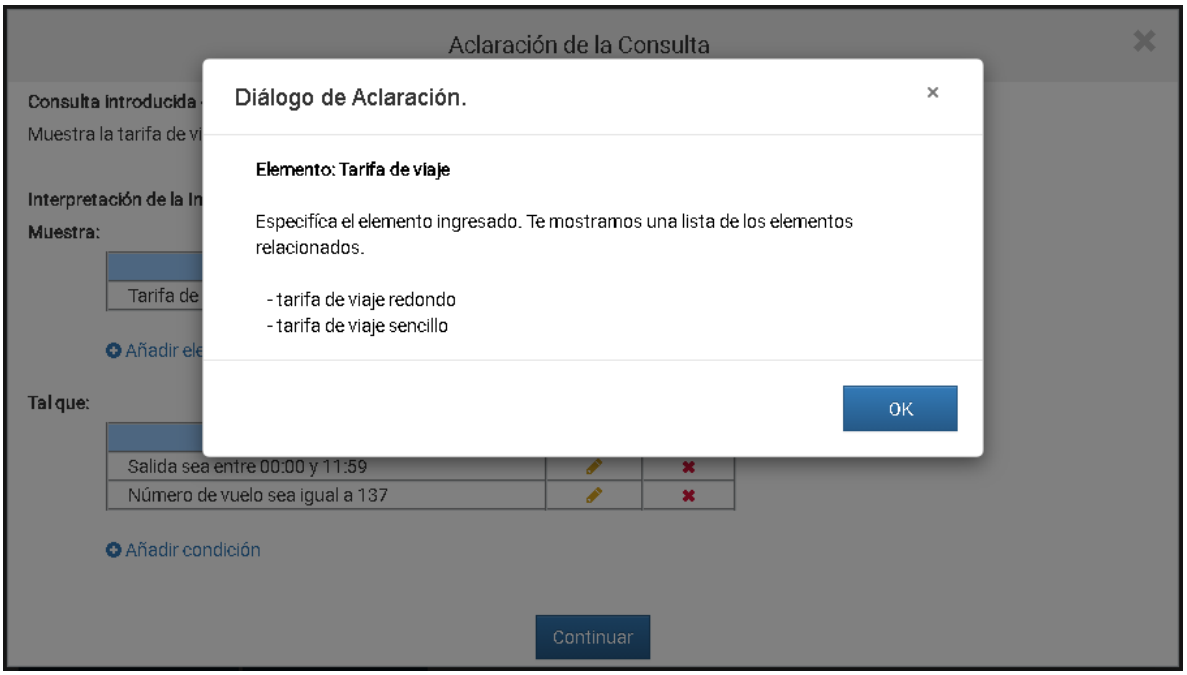


Figura 4.5 Diálogo de aclaración: Específica el elemento del apartado *Muestra*.

Por otra parte, si la interfaz no logra asociar el elemento ingresado con alguna columna o simplemente, éste no existe en la base de datos, la interfaz muestra un diálogo donde se proporciona un listado de sugerencias de elementos con relación a la BD consultada, como se muestra en la Figura 4.6.

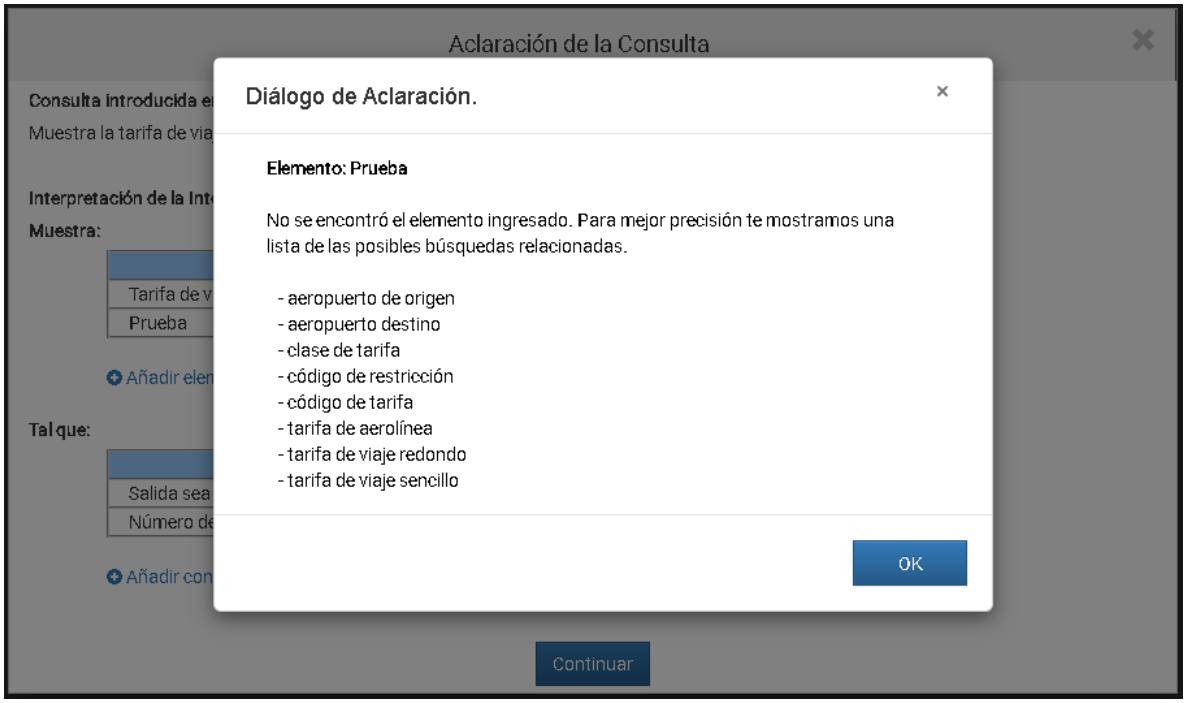


Figura 4.6 Diálogo de aclaración: No se encontró el elemento del apartado *Muestra*.

En el Algoritmo 4.5 se presenta el pseudocódigo general para verificar y validar cada uno de los elementos *columna* del apartado *Muestra,* el cual se codificó en la clase *AdminDialogo.java.* En el caso de la verificación inicial para el ingreso de al menos un elemento a mostrar, la confirmación de eliminación y otras funciones para eliminar, cambiar y agregar, se encuentran codificadas en *eventos_aclaracion.js.*

Algoritmo 4.5 Pseudocódigo para validar columna

```
1   LE // Lista de elementos
2   Cad // Cadena
3   Msg // Diálogo
4   LC // Lista del descriptor columna
5   SQL_DIS // Consulta del DIS
6   if esColumna(cad)
7       LE ← getColumna(cad)
8   else
9       for i = 0, …, cad-1 do
10          tokens ← cad.token()
11      endfor
12      while column.next(SQL_DIS(tokens))
13          LE ← getColumna(tokens)
14          LC ← getDescriptorColumn(tokens)
15      endwhile
16
17      if size(LC) > 1
18         Msg ← "Especificar columna" + LC
19      endif
20      if isEmpty(LC)
21          while column.next(SQL_DIS(*))
22              posibles_búsquedas ← getColumna()
23          endwhile
24          Msg ← "Columna inexistente" + posibles_búsquedas
25      endif
    endif
```

4.2.3 Validación del apartado *Tal que*

En el proceso de validación de condiciones del apartado *Tal que* no se verifica que exista al menos una condición a diferencia de la verificación inicial en el apartado de *Muestra*, ya que para este caso si se puede continuar con el proceso de la aclaración.

En el caso de que se encuentre una o más condiciones de búsqueda ingresadas en el apartado, se hace un recorrido por cada una, al cual inicialmente se le realiza una limpieza de los datos a procesar, para posteriormente etiquetar cada palabra como columna, tabla, palabra reservada (comando, clausula, operador lógico, operador de comparación) o valor de búsqueda, de acuerdo al diccionario de información semántica. Posteriormente todas las palabras que fueron marcadas con la misma etiqueta de *columna* y *operador comparación* son enviadas a un método para seleccionar una columna y un operador de comparación por cada condición de búsqueda (Algoritmo 4.6).

Algoritmo 4.6 Pseudocódigo del etiquetado de condición

```
1   L // Lista de etiquetas
2   Cad // Cadena
3   Msg // Diálogo
4   clean(Cad)
5   for i = 0, …, cad-1 do
6       token ← cad.token()
7       if esColumna(token)
8          L ← etiquetaColumna(token)
9       else
10          if esPalabraReservada(token)
11             L ← etiquetaPalabraReservada(token)
12          else
13             if esTabla(token)==false and esPalabraInutil(token) ==false
14                L ← etiquetaValorBusqueda(token)
15             endif
16          endif
17      endif
18   endfor
19
20  for i = 0, …, L-1 do
21      columnas ← etiquetaColumna(Li)
22      operador_comp ← etiquetaOperadorComp (Li)
23      valores ← etiquetaValor(Li)
24  endfor
25
26  columna_final ← getColumna(columnas)
27  operador_final ← getColumna(operador_comp)
28  valor_final ← getValorAlias_Impreciso(valores)
```

Igual que en el apartado *Muestra*, se verifica que el componente *columna* esté asociado a una sola columna de la BD a consultar, por lo tanto, si la interfaz identifica más de una asociación con otras columnas se muestra un diálogo de aclaración con el mensaje que se muestra en la Figura 4.7, donde se indica la condición a aclarar, con la posible opción de visualizar un listado de sugerencias con relación al componente ingresado en la condición, de este modo se pretende que el usuario precise la condición de la consulta.

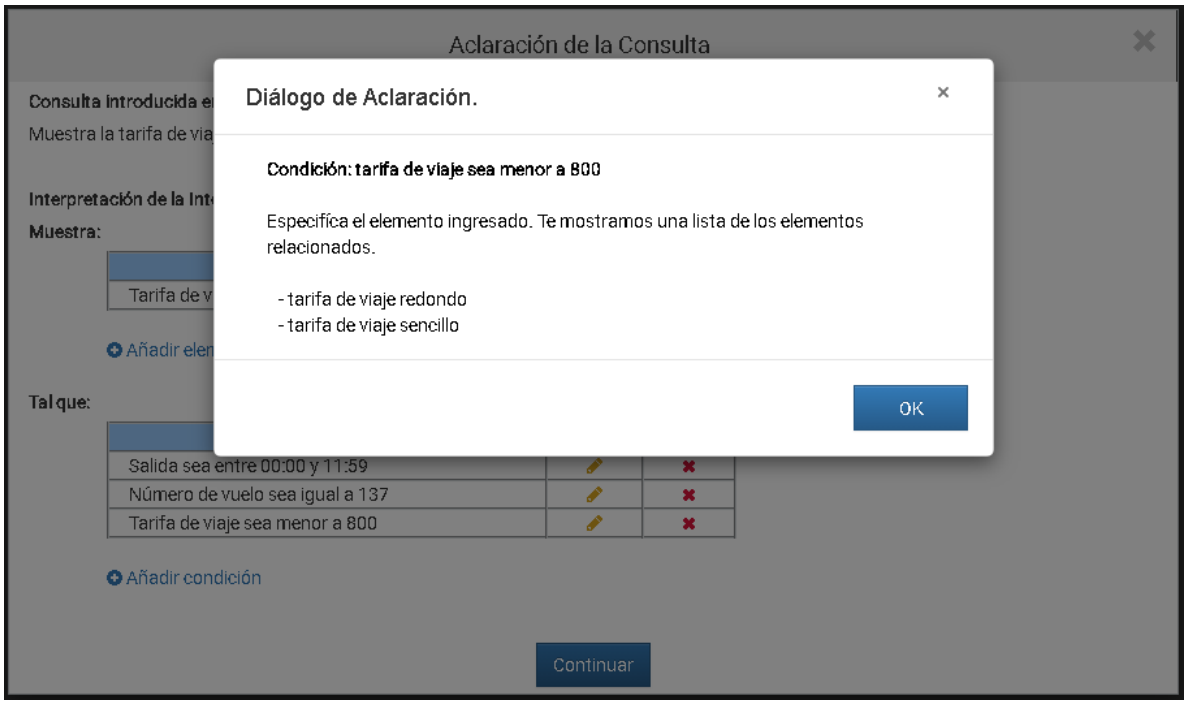


Figura 4.7 Diálogo de aclaración: Específica el elemento del apartado *Tal que*.

En caso de que la interfaz no logre asociar algún componente *columna* ingresado, debido a que el ingreso de palabras es libre, se le notifica al usuario que no se pudo crear la condición por una de las siguientes tres razones posibles (Figura 4.8):

1. que la interfaz no haya logrado identificar el componente *columna* en la condición, donde también se le indica con un ejemplo la estructura preferente;
2. que el componente *columna* ingresado no exista en la base de datos;
3. que el usuario haya ingresado más de un componente *columna* en la condición sobre el mismo recuadro.

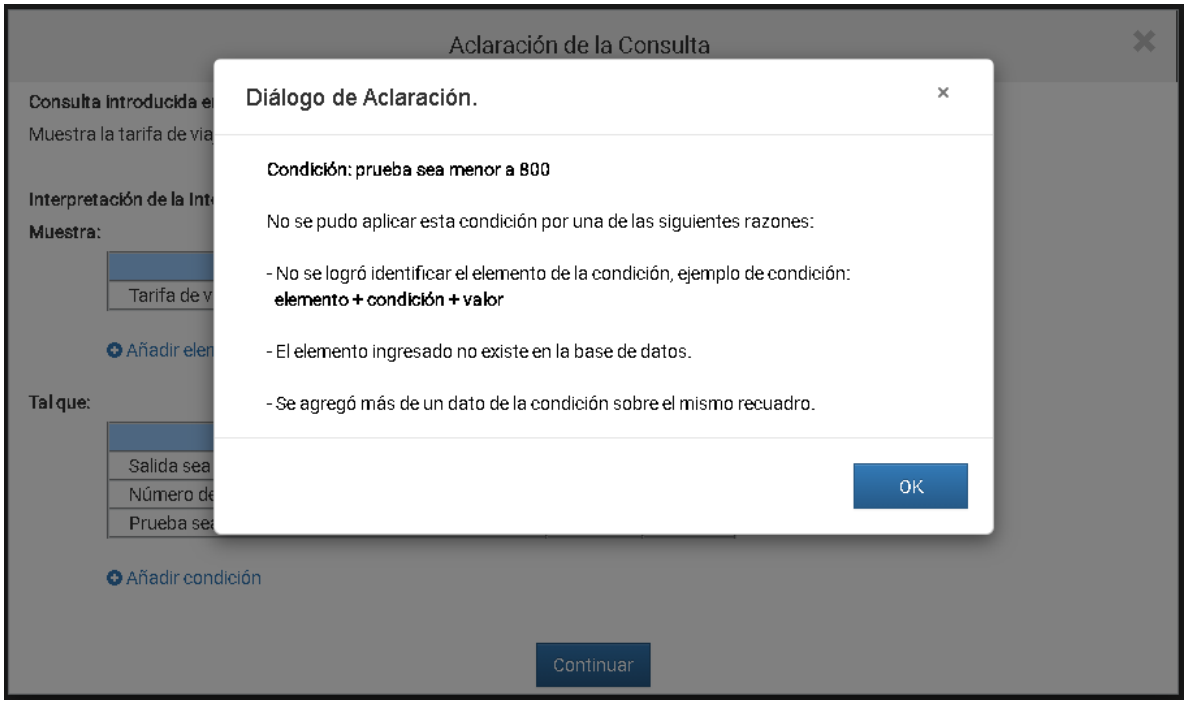


Figura 4.8 Diálogo de aclaración: No se encontró el elemento del apartado *Tal que*.

Para el caso del *operador de comparación*, también se verifica que en la estructura de la condición se haya ingresado el componente *operador*, el cual hace referencia a la comparación del componente *columna* con el valor de búsqueda, de modo que la interfaz sea capaz de interpretarlo.

A continuación se describen los diferentes casos de verificación del componente *operador:*

- Uno de los casos que puede presentarse es cuando la comparación en la condición de búsqueda no es específica. Por ejemplo, en una condición:

  El número de vuelo sea 137

No queda claro si éste debe ser igual, mayor, menor, diferente o entre otras comparaciones a 137. Para éste caso se presenta el siguiente diálogo de aclaración de la Figura 4.9, dónde se enlista todas las posibles comparaciones que el usuario puede ingresar.

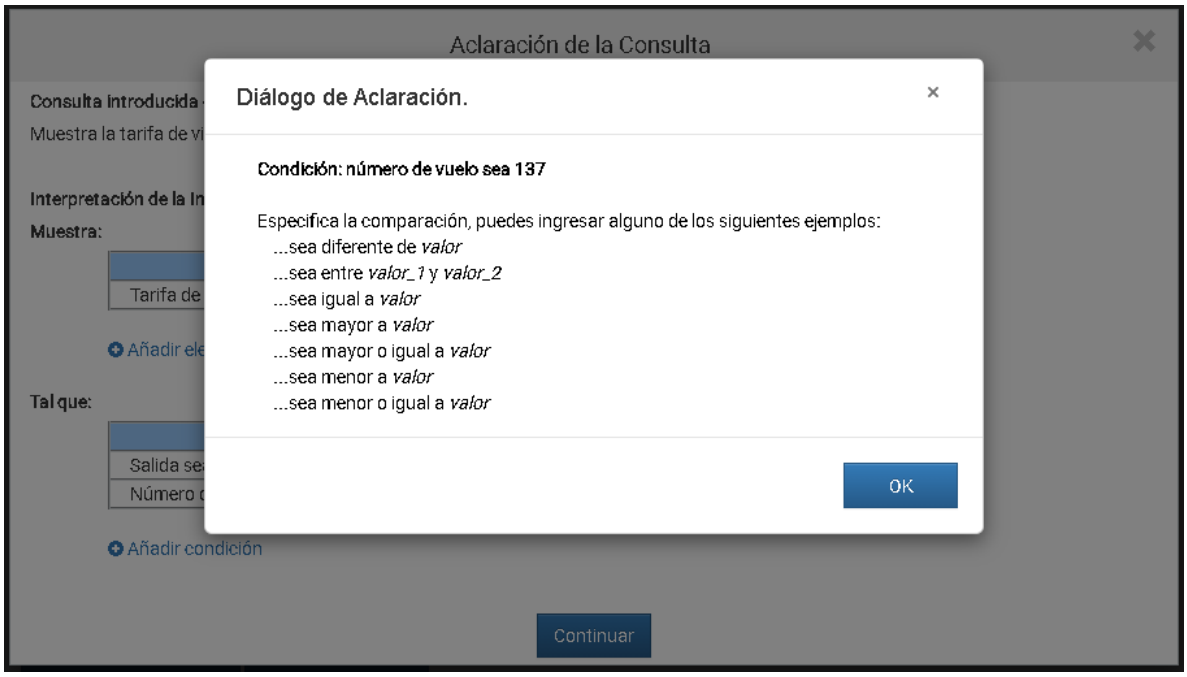


Figura 4.9 Diálogo de aclaración: Especifica la comparación.

En cambio, un ejemplo de una comparación específica de una condición común es la siguiente:

El número de vuelo sea igual a 137

Debido a que en este caso se cumple con la regla de condición siguiente:

columna + operador_comparación + valor_búsqueda

No se muestra ningún diálogo de aclaración y únicamente se continúa con el proceso.

- Otro caso que se verifica es en la comparación del rango de valores (*between*) no específico. Considerando como ejemplo la condición siguiente:

  Salida sea entre las 00:00 y 11:59

  Si la oración cumple con la regla de condición:

  columna + operador_between + valor_de_búsqueda_1 + operador_lógico + valor_de_búsqueda_2

Se continua con el proceso, por lo contrario si un valor de búsqueda no es ingresado o si la cantidad de valores ingresados son excedidos para cumplir con un rango de comparación, se notifica al usuario mediante un diálogo de aclaración solicitándole que especifique el rango de condición e indicado con un ejemplo como se muestra en la Figura 4.10.

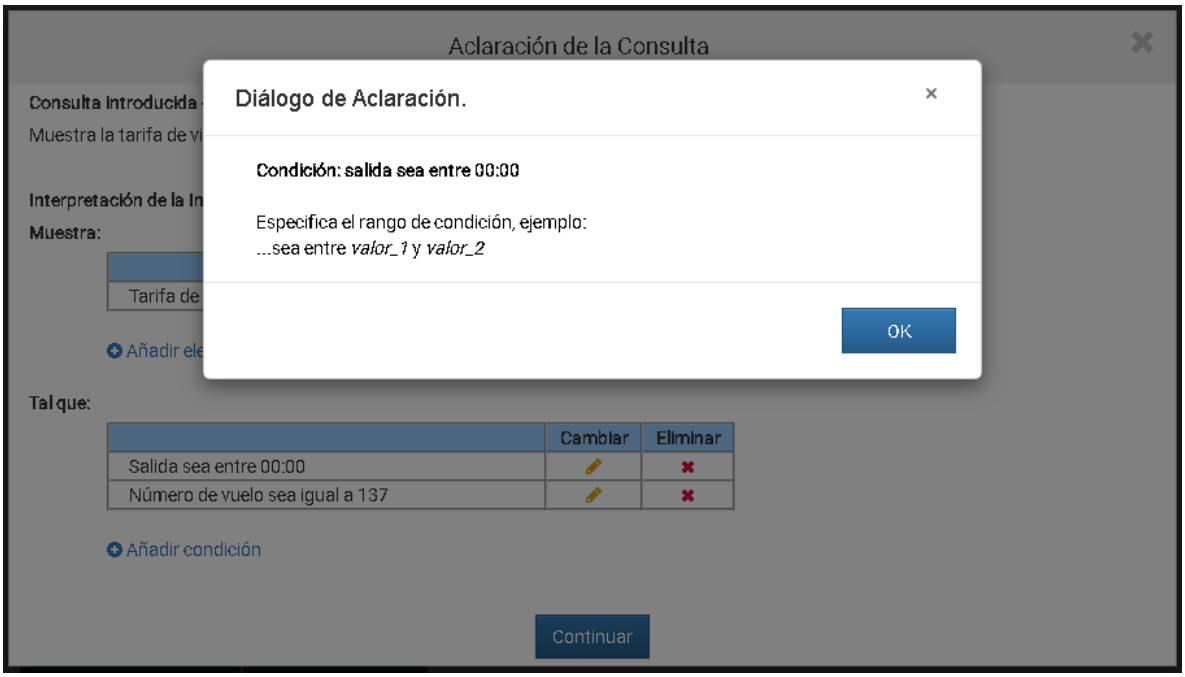


Figura 4.10 Diálogo de aclaración: Especifica el rango de condición.

- Otro caso es cuando no se logra identificar el componente *operador* en la condición de búsqueda. Por ejemplo, en una condición:

  Número de vuelo 137

  No se indica el tipo de comparación que desea que se aplique a la condición, por lo tanto, la interfaz no sería capaz de aplicar esta condición, sin embargo se le notifica al usuario de este caso, donde también se proporciona un listado de las posibles comparaciones a utilizar, como se muestra en la Figura 4.11.

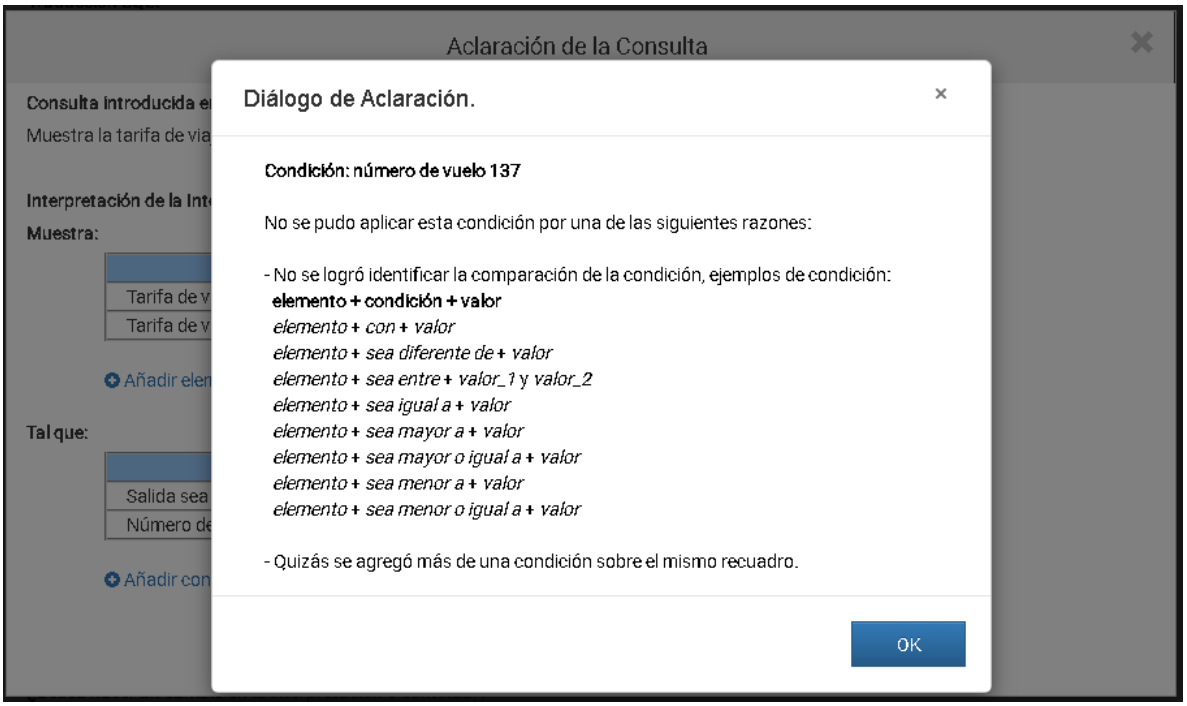


Figura 4.11. Diálogo de aclaración: No se encontró la comparación de la condición.

Por último, se verifica también que exista el componente *valor de búsqueda*, ya que sin este dato la interfaz no sería capaz de aplicar la condición en la consulta. En el caso de que la interfaz no logre identificar este componente, se muestra un diálogo de aclaración como el de la Figura 4.12.

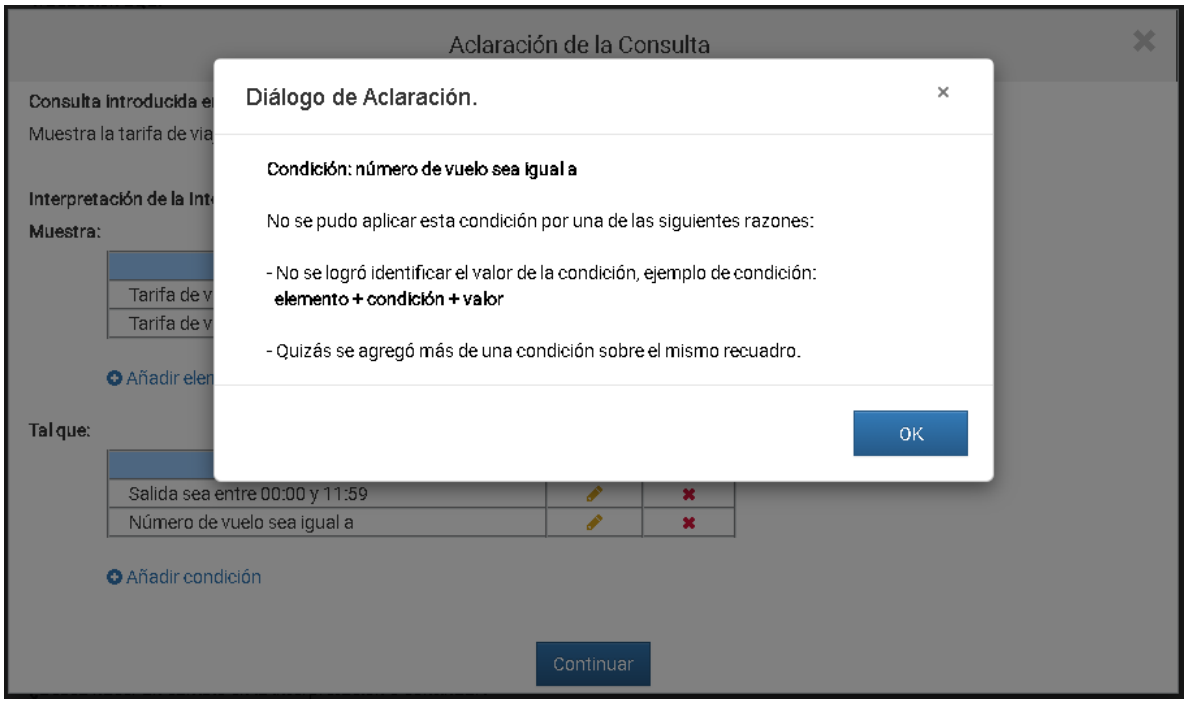


Figura 4.12. Diálogo de aclaración: No se encontró el valor de búsqueda.

En el Algoritmo 4.7 se presenta el pseudocódigo general para verificar y validar cada una de las condiciones de búsqueda del apartado *Tal que*, el cual se codificó en la clase *AdminDialogo.java.* Para el caso de la confirmación de eliminación y otras funciones para eliminar, cambiar y agregar condiciones de búsqueda, se encuentran codificadas en *eventos_aclaracion.js.*

Algoritmo 4.7 Pseudocódigo para validar condición

```
1   Msg // Diálogo
2   LC // Lista de condiciones
3   if  operador_final != ""
4       if  columna_final != ""
5           if  valor_final != ""
6               if operadorUndefine(operador_final)
7                   Msg ← "Especifica la comparación"
8               else
9                   if operadorBetween(operador_final)
10                      if valor_final.length==2
11                          condición ← columna_final + operador_final + valor_final_1 + AND +
12                      valor_final_2
13                      else
14                          Msg ← "Especifica el rango de condición"
15                      endif
16                  else
17                      if operadorComp(operador_final)
18                          condición ← columna_final + operador_final + valor_final
19                      else
20                          Msg ← "La condición no cumple con algún dato requerido"
21                      endif
22                  end if
23              endif
24              LC ← condición
25          else
26              Msg ← "No se identificó el valor de búsqueda de la condición"
27          endif
28      else
29          Msg ← "No se identificó la columna de la condición"
30      endif
31  else
32      Msg ← "No se identificó la comparación de la condición"
33  endfor
```

Por último, una vez completado y validado el proceso de diálogo para la aclaración de la consulta, la interfaz envía la nueva instrucción SQL obtenida al proceso de la paráfrasis y al SABD, la cual será transparente para el usuario, para extraer la nueva interpretación e información de la BD seleccionada, con el fin de mostrar al usuario el resultado solicitado como se muestra en la Figura 4.13.

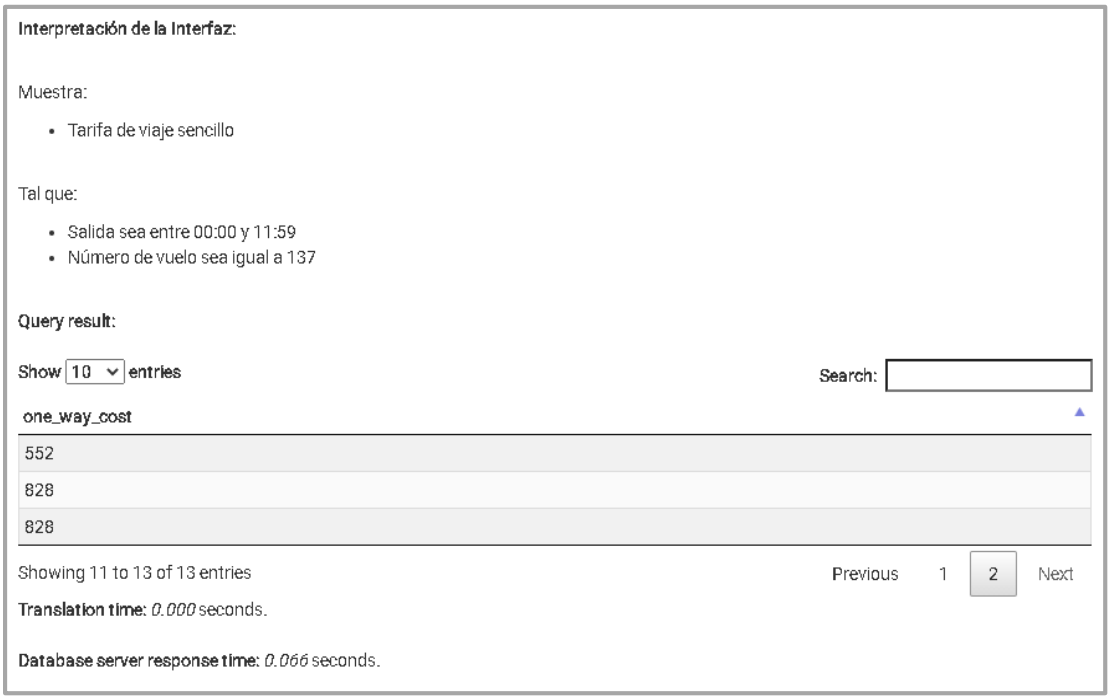


Figura 4.13. Interpretación y resultados de la consulta.

Capítulo 5
Pruebas de la ILNBD

En este capítulo se presentan las características del hardware y software del equipo utilizado para la ejecución de las pruebas funcionales. También se presentan las pruebas realizadas a la ILNBD para web posterior a la implementación descrita en el capítulo anterior, se ejemplifica el uso del módulo para la aclaración de consultas y se muestran los resultados obtenidos de las pruebas realizadas con un subconjunto del corpus de ATIS y Geobase.

5.1 Descripción del hardware y software del equipo

En la Tabla 5.1 se muestran las especificaciones del hardware que se utilizó para la ejecución de las pruebas.

Tabla 5.1 Características del hardware.

Características	Especificaciones
Procesador	AMD E2- 7110 @ 1.80 GHz
Memoria	4 GB

En la Tabla 5.2 se muestran las especificaciones del software que se utilizó para la ejecución de las pruebas.

Tabla 5.2 Características del software.

Características	Especificaciones
Sistema Operativo	Windows 10 Home Single de 64 bits
Entorno	NetBeans IDE 8.2
Lenguaje	Java 1.8
Servidor Web	Apache Tomcat 9.0.8
SABD	PostgreSQL 10

5.2 Pruebas funcionales de la ILNBD

Las pruebas funcionales que se realizaron a la ILNBD con el módulo de aclaración de consulta consistieron en ejecutar un subconjunto del corpus de la base de datos utilizada para el desarrollo de este proyecto.

A continuación se muestran algunos ejemplos de aclaración de consulta:

Ejemplo 1

Tipo de consulta:

Consulta con elipsis semántica por falta de información de tablas o columnas.

Consulta en lenguaje natural:

¿Puede decirme la tarifa del vuelo 16?

Consulta SQL obtenida del traductor:

```
SELECT fare.rnd_trip_cost, 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.flight_number = 16;
```

Interpretación (Paráfrasis):

Muestra**:**

- Tarifa de viaje redondo
- Tarifa de viaje sencillo

Tal que**:**

- Número de vuelo sea igual a 16

Aclaración con el administrador de diálogo:

a. Eliminar el elemento no deseado.
b. Cambiar el elemento solicitado.
c. Agregar un nuevo elemento.

En este ejemplo se elimina del elemento "Tarifa de viaje redondo" del apartado *Muestra*.

Nueva consulta SQL:

```
SELECT fare.one_way_cost
FROM fare, flight, flight_fare
WHERE flight_fare.flight_code = flight.flight_code AND fare.fare_code =
flight_fare.fare_code AND flight.flight_number = 16;
```

Nota: Se eliminó la selección de la columna fare.rnd_trip_cost (Tarifa de viaje redondo) del SELECT.

En este ejemplo se representa un tipo de consulta con elipsis semántica por falta de información (en este caso para la columna) que es necesaria para que el traductor logre identificar con precisión la información a mostrar. Por medio de la paráfrasis el usuario puede visualizar lo que la interfaz interpretó y la información que se mostrará como resultado. En este ejemplo el componente léxico "tarifa" hace referencia a dos posibles columnas (Tarifa de viaje redondo y Tarifa de viaje sencillo), sin embargo, para la interfaz no queda claro cuál de estas dos columnas debería mostrar. Para tratar la aclaración de esta consulta, el usuario puede elegir cambiar la interpretación de la interfaz, el cual muestra un administrador de diálogo que tiene como opciones; eliminar elemento no deseado, cambiar el mismo o agregar un nuevo elemento al componente SELECT.

Ejemplo 2

Tipo de consulta:

Consulta con un valor de búsqueda de alias y con un valor sin formato.

Consulta en lenguaje natural:

Enumere todos los vuelos que salen después del mediodía y llegan antes de las 700.

Consulta SQL obtenida del traductor:

```
SELECT flight.flight_number
FROM flight
WHERE flight.arrival_time<700 AND flight.departure_time>1200;
```

Interpretación (Paráfrasis):

Muestra**:**

- Número de vuelo

Tal que**:**

- Llegada sea menor a 07:00
- Salida sea mayor a 12:00

Aclaración con el administrador de diálogo:

a. Eliminar la condición de búsqueda que no se desea aplicar.
b. Cambiar la condición de búsqueda solicitada.
c. Agregar una nueva condición de búsqueda.

En este ejemplo se hace el cambio de la condición "Llegada sea menor a 07:00" por "Llegada sea menor a 19:00" en el apartado de *Tal que*.

Nueva consulta SQL:

```
SELECT flight.flight_number
```

FROM flight
WHERE flight.arrival_time < '1900' AND flight.departure_time > '1200';

Nota: Se cambió el valor de 700 por 1900 en la condición de la columna flight.arrival_time del WHERE.

En este segundo ejemplo se representa un tipo de consulta que contiene un valor de búsqueda de alias, el componente léxico “mediodía”, el cual esta interfaz es capaz de traducirla a un valor válido (1200) para realizar la consulta, así también contiene un valor sin formato, el componente léxico “700”, el cual ya es un valor válido para realizar la consulta. Cabe aclarar que estos valores hacen referencia a un horario, sin embargo, al momento de realizar la paráfrasis estos valores son basados en el horario de 24 horas, por lo cual si se requiere que el valor “700” haga referencia a un horario de la tarde (p.m.) se tuvo que haber ingresado el valor de “1900”. Para tratar la aclaración de esta consulta, el usuario puede elegir cambiar la interpretación de la interfaz mediante el administrador de diálogo, en el cual es posible cambiar el valor de búsqueda de la consulta, en este caso se puede especificar el valor con un formato de 24 horas (por ejemplo, 19:00).

Por otra parte, si el usuario desconoce esta información, de que el formato está basado en el horario de 24 horas o simplemente desconoce el formato, podría generar confusión en la interpretación, debido a que “07:00” podría llegar a pensarse que hace referencia a la mañana o la tarde. Para lo cual en el administrador de diálogo también se puede especificar el valor con un formato de 12 horas (por ejemplo, 07:00 p.m.) para la aclaración.

Además de cambiar la condición de búsqueda solicitada en el administrador de diálogo, también se puede eliminar la condición de búsqueda que no se desee aplicar o en su defecto, agregar una nueva.

5.3 Resultados

Los experimentos que se realizaron al módulo de aclaración de consulta de la ILNBD para web consistieron en ejecutar un subconjunto del corpus de la base de datos ATIS (Apéndice A) que contiene información de viajes aéreos, y Geobase (Apéndice B) con información geográfica de los Estados Unidos de América, los cuales se obtuvieron de la tesis de Aguirre. En este experimento las consultas fueron traducidas al idioma español.

Para la evaluación del desempeño se utilizó la métrica *recall*, la cual es el porcentaje de consultas respondidas correctamente con respecto al número de consultas ingresadas en la interfaz (Pazos, 2013).

$$\text{recall} = \frac{\text{número total de consultas correctas}}{\text{número total de consultas}} \times 100$$

Los resultados de la prueba descrita para el corpus de la base de datos ATIS que se presentan en la Tabla 5.3, como se puede apreciar en esta ILNBD para web, configurado con un wizard y utilizando el administrador de diálogo, se obtuvo un resultado de 66 consultas contestadas correctamente de un total de 70 consultas, en comparativa con la versión anterior, se logró un incremento en el porcentaje de éxito de un 9% para este corpus de consultas.

Tabla 5.3 Resultados obtenidos con el corpus de la base de datos ATIS.

ILNBD	BD	Afinación con Wizard	Admin. Diálogo	Total de consultas	Consultas correctas	Porcentaje
ILNBD Web [González,2018]	ATIS	✓		70	60	85%
ILNBD	ATIS	✓	✓	70	66	94%

En la Tabla 5.4 se muestran los resultados obtenidos con el corpus de la base de datos Geobase, como se puede apreciar en esta ILNBD para web, configurado con un wizard y utilizando el administrador de diálogo, se obtuvo un resultado de 38 consultas contestadas correctamente de un total de 41 consultas, en comparativa con la versión anterior, se logró un incremento en el porcentaje de éxito de un 5% para este corpus de consultas.

Tabla 5.4 Resultados obtenidos con el corpus de la base de datos Geobase.

ILNBD	BD	Afinación con Wizard	Admin. Diálogo	Total de consultas	Consultas Correctas	Porcentaje
ILNBD Web [González,2018]	Geobase	✓		41	36	87%
ILNBD	Geobase	✓	✓	41	38	92%

La consultas que no fueron respondidas correctamente por la ILNBD, se debe a consultas no tratadas por la interfaz, por ejemplo, consultas que solicitan condiciones de búsqueda semejantes, es decir, hacen referencia a una misma columna pero con diferente valor de búsqueda. Una de estas consultas es *Dame una lista de todos los vuelos de PHL a ATL y de ATL a DFW.* La consulta crea cuatro condiciones de búsqueda, de las cuales hace referencia dos veces a la columna *from_airport* y *to_airport* con diferentes valores de búsqueda, en este caso la interfaz asocia con el operador lógico AND todas las condiciones de búsqueda de la cláusula WHERE, en lugar de usar el operador lógico OR para filtrar sobre una misma columna los diferentes valores de búsqueda, esto implica que al ejecutar la consulta SQL el resultado sea diferente al deseado.

Capítulo 6
Conclusiones

En este capítulo se presentan las conclusiones referentes a este proyecto de tesis y los posibles trabajos futuros.

6.1Conclusiones

El desarrollo de este proyecto de tesis tuvo como propósito implementar un módulo de aclaración de consultas para una ILNBD, éste consiste en aclarar la consulta original, la cual para poder consultar una base de datos ésta se traduce a un lenguaje de consulta SQL por la interfaz, el cual para un usuario que carece de conocimientos de un lenguaje de consulta, sería bastante difícil de interpretar. Por tanto, se propuso dividir (SELECT y WHERE) y mostrar previamente en dos apartados (*Muestra* y *Tal que*) la paráfrasis de la consulta SQL, donde también se solicita al usuario cambiar la interpretación (con ayuda de un administrador de diálogo) o continuar con el proceso.

La paráfrasis ayuda al usuario identificar el(los) elemento(s) que se mostrarán y la(s) condición(es) de búsqueda que se deben cumplir en la información solicitada. Así también se clarifican los valores de búsqueda alias e imprecisos que la interfaz interpreta de acuerdo al diccionario de información semántica.

Por otra parte, el administrador de diálogo habilita las opciones de eliminar, cambiar o agregar el(los) elemento(s) que se mostrarán y/o la(s) condición(es) de búsqueda, en la cual en caso de ser necesario se notifica al usuario mediante diálogos hacer la aclaración, como por ejemplo, en la falta de especificación del nombre de una columna o la inexistencia de la columna en la base de datos a consultar, otro de los casos considerados es la falta de algún componente para formar una condición de búsqueda, esta aclaración también facilita de manera interna a la interfaz crear una nueva consulta SQL con mejor precisión.

De este modo se amplía la funcionalidad de la ILNBD para web que se desarrolla en el ITCM para mejorar la interpretación de lo que los usuarios consultan; así también, se aumenta el porcentaje de respuestas certeras y confiables, reduciendo el margen de error y evitando que el usuario tenga que aprender un lenguaje formal como el SQL.

6.2 Trabajos futuros

El desarrollo de la ILNBD para web desarrollada en el ITCM, en particular el módulo de aclaración, puede mejorar en el desarrollo de procesos de diálogo para aclarar problemas que presentan las consultas que involucran funciones de agregación, agrupación y de orden.

Así también en el desarrollo de diálogos para la aclaración de las consultas que involucran repuestas booleanas (Si/No, Verdadero/Falso, 1/0), que aun cuando estas pueden ser tratadas con el administrador de diálogo, dependen mucho del conocimiento del tipo de dato que guardan las columnas, lo cual es imposible para un usuario común que desconoce la estructura de la base de datos.

Apéndice

Apéndice A. Corpus de consultas para la BD ATIS.

1 Can you tell me the fare for flight 16?
2 Cost of flight of flight number 144165.
3 Give me a list of all aircraft types.
4 Give me a listing of all aircraft available, the capacity and the weight.
5 Give me a listing of all equipment sizes and speed.
6 I'd like the airfares for flight 11 and 12.
7 I'd like the price list of flight number 3.
8 List all the flight restrictions.
9 List all the types of ground transportation.
10 List equipment capacity.
11 List round-trip fares.
12 List the airplanes.
13 List the categories of aircraft.
14 List the cities.
15 List the class codes of flights.
16 List the classes of service of flights.
17 List the fare for flight number 9.
18 List the transport descriptions.
19 List the type of airplanes for flights from DFW to BWI.
20 Name all the airports.
21 Show me the cost of flight 9.
22 Please list all flights.
23 Please show me the business fare class cost for flight number 1.
24 Round-trip airfare for the flight from ATL.
25 Let me see the flights from OAK to BWI arriving before noon.
26 List fares for all flights leaving after 1200 from BOS to BWI.
27 Please display the trips that are only for SJC departures.
28 Could I see the the airline and flight number that I would be leaving out of SFO?
29 Give me a list of all flights from DFW to BOS that arrive before 700.
30 How much do flights number 1, 2, 3, 4 and 5 cost?
31 How much do the flights from ATL to SFO cost?
32 How much does flight number 90 and 888 from DEN to DFW cost?
33 How much does it cost to fly from BOS to OAK one-way?
34 How much is a round-trip fare from BOS to DFW?

35 I need flights that arrive before noon.
36 List all flights leaving after noon and arriving before 700.
37 List all the flights leaving from DEN to PIT after 500, and list the fares.
38 List only flights arriving before 700.
39 List only flights leaving from SFO.
40 List the flights that depart from SJC.
41 Please just show me the flights leaving SFO.
42 Please list only economy class flights leaving after noon.
43 Please show me flights that leave after noon.
44 Please show the airlines which fly from DFW to DEN.
45 Please display departure time.
46 Can I have a listing of all flights from ATL to BOS?
47 Give me a listing of flights from DEN to SFO.
48 List all flights from DEN to PIT and list the fares.
49 May I see flights from SFO to LAC?
50 Show all flights and fares from DFW to DEN.
51 Flights from SFO to DFW.
52 From OAK to BOS, what is the fare?
53 Give all flights from DFW to BOS to DEN.
54 Give me a listing from all flights from PHL to ATL and from ATL to DFW.
55 Give me the fare on class Q from DFW to ATL.
56 Find the cost of a one-way flight from PIT to OAK.
57 Give me a round-trip fare from ATL to BWI.
58 List all fare prices for airlines from DFW to DEN.
59 List all flights from OAK to SFO, showing the prices.
60 List flights from ATL to SFO.
61 List the round-trip fares from DFW to ATL.
62 May I have a list of fares from ATL to BOS?
63 One-way airfare from WAS to ATL.
64 What afternoon flights are available from WAS to BOS with meals?
65 Give me a list of flights from PHL to BWI in the morning.
66 Give me the prices of all the flights from DFW to BOS in the morning.
67 List afternoon flights from ATL to SFO.
68 List all the flights from DFW to BOS in the morning.
69 Please show the cost of the morning flights from DFW to DEN.
70 Please show the list of flights from DFW to DEN in the morning and show their cost.

Apéndice B. Corpus de consultas Geoquery250 para la BD Geobase.

1. Which rivers run through states bordering new mexico?
2. What is the highest point in montana?
3. What is the most populated state bordering oklahoma?
4. Through Which states does the mississippi run?
5. What is the longest river?
6. How long is the mississippi?
7. Which state has the smallest population density?
8. What is the area of wisconsin?
9. What is the lowest point of the state with the largest area?
10. What is the longest river in mississippi?
11. What states border montana?
12. What states border new jersey?
13. Which state has the longest river?
14. Name the rivers in arkansas.
15. Which states have points higher than the highest point in colorado?
16. How many people live in the capital of texas?
17. How long is the delaware river?
18. What is the smallest city in the usa?
19. What states border georgia?
20. What is the smallest state by area?
21. How long is the mississippi river?
22. What states border delaware?
23. What is the shortest river in the usa?
24. What states have cities named plano?
25. How many rivers does colorado have?
26. What is the biggest city in georgia?
27. What states border hawaii?
28. What is the capital of the state with the highest point?
29. What state has the highest population?
30. What is the capital of maine?
31. Which state borders florida?
32. What state has highest elevation?
33. What rivers run through the states that border the state with the capital atlanta?
34. What is the biggest city in oregon?
35. What is the lowest point of the us?
36. Which state borders hawaii?
37. What are the major cities in ohio?
38. What is the population of springfield missouri?

39 How many people live in california?
40 Where is the highest point in montana?
41 What are the major cities in alaska?
42 What are the major cities in kansas?
43 Which state has the highest point?
44 What states border florida?
45 What states does the ohio river go through?
46 What is the largest city in minnesota by population?
47 How many rivers are there in idaho?
48 How high is the highest point in montana?
49 What is the lowest point in california?
50 What is the capital of georgia?
51 How big is texas?
52 What is the highest point in nevada in meters?
53 How many people live in minneapolis minnesota?
54 What is the area of maine?
55 What is the lowest point in oregon?
56 What state has the city flint?
57 Give me the largest state?
58 How many states does the colorado river run through?
59 What is the area of south carolina?
60 Which state has the highest elevation?
61 How large is alaska?
62 How many citizens live in california?
63 What is the biggest city in wyoming?
64 Which states border south dakota?
65 What state has the largest population density?
66 What is the population of utah?
67 How many people live in rhode island?
68 What is the population of new york city?
69 Which states border texas?
70 What is the population of seattle washington?
71 What is the highest point in colorado?
72 How large is the largest city in alaska?
73 What is the longest river in the us?
74 How many states does the mississippi river run through?
75 What are the high points of states surrounding mississippi?
76 What is the highest point of the usa?
77 What is the largest river in washington state?
78 What is the population of illinois?
79 Which state borders the most states?

80 Which rivers flow through alaska?
81 What city has the most people?
82 Which states does the mississippi run through?
83 What is the capital of washington?
84 What is the smallest city in the us?
85 What are the major cities in texas?
86 Which state has the highest population density?
87 What state contains the highest point in the us?
88 What states does the delaware river run through?
89 Which states capital city is the largest?
90 How many citizens in alabama?
91 What is the highest point in states bordering georgia?
92 What rivers are in utah?
93 What is the area of the largest state?
94 What are all the rivers in texas?
95 What is the population density of wyoming?
96 What is the capital of new jersey?
97 What is the lowest point in nebraska in meters?
98 What major rivers run through illinois?
99 What is the capital of new hampshire?
100 What is the lowest point in massachusetts?
101 What is the largest city in states that border california?
102 What states border indiana?
103 Where is the lowest spot in iowa?
104 How many square kilometers in the us?
105 What is the highest point in rhode island?
106 What are the major cities in rhode island?
107 What states border arkansas?
108 Where is the lowest point in the us?
109 Rivers in new york?
110 What is the population density of maine?
111 What is the lowest point in the state of california?
112 What is the highest point in the us?
113 How long is the colorado river?
114 How long is the north platte river?
115 How large is texas?
116 Which states border colorado?
117 What is the lowest point in louisiana?
118 What is the population of dallas?
119 What is the population of tempe arizona?
120 How many rivers in washington?

121 What is the shortest river in the us?
122 What are the major cities of texas?
123 How many people live in kalamazoo?
124 How many rivers does alaska have?
125 What rivers run through colorado?
126 What is the length of the colorado river?
127 What is the state with the lowest population?
128 What states border rhode island?
129 How many rivers are in colorado?
130 What is the total population of the states that border texas?
131 What is the length of the mississippi river?
132 What is the population of oregon?
133 How many cities are there in the us?
134 What is the area of alaska?
135 How many people live in spokane washington?
136 What is the combined population of all 50 states?
137 What state has the capital salem?
138 How high is the highest point in america?
139 What is the biggest city in the us?
140 What is the smallest city in alaska?
141 How long is the shortest river in the usa?
142 What states have cities named dallas?
143 What is the biggest river in illinois?
144 What is the capital of iowa?
145 What is the highest point in iowa?
146 What is the population density of texas?
147 What is the longest river in florida?
148 What is the population of hawaii?
149 What is the smallest city in washington?
150 What are the major cities in oklahoma?
151 What state is des moines located in?
152 What is the highest point in the country?
153 What state borders michigan?
154 What states border new hampshire?
155 What is the lowest point in the united states?
156 How long is the rio grande river?
157 What are the major rivers in ohio?
158 What is the capital of north dakota?
159 What is the largest city in rhode island?
160 What is the population of the capital of the smallest state?
161 What is the most populous state?

162 What is the largest city in wisconsin?
163 What is the population of the major cities in wisconsin?
164 Give me the cities in virginia?
165 Which states have cities named austin?
166 What state is columbus the capital of?
167 What is the city with the smallest population?
168 What states does the missouri run through?
169 What is the longest river in the united states?
170 How many cities are in montana?
171 What is the highest elevation in new mexico?
172 How long is the missouri river?
173 What capital is the largest in the us?
174 What is the population of south dakota?
175 How many people live in new york?
176 What is the population of san antonio?
177 What are the major cities in california?
178 What state has the greatest population density?
179 Which river runs through the most states?
180 Which states does the missouri river run through?
181 Which state has the highest peak in the country?
182 What is the biggest city in arizona?
183 What is the lowest point in the state of texas?
184 Which state is the city denver located in?
185 What is the lowest point in arkansas?
186 What is the biggest city in texas?
187 What is the biggest city in the usa?
188 Which state has the largest city?
189 How many rivers are in new york?
190 What is the lowest point in texas?
191 Which states border kentucky?
192 Which state borders most states?
193 How many major cities are in florida?
194 What are the major cities in wyoming?
195 What is the highest point in the usa?
196 What is the population density of the smallest state?
197 Name all the rivers in colorado?
198 What is the capital of vermont?
199 What is the population of tucson?
200 What is the highest mountain in the us?
201 What is the capital of utah?
202 How long is the ohio river?

203 What rivers do not run through tennessee?
204 What is the highest point in wyoming?
205 Which states does the mississippi river run through?
206 What states capital is dover?
207 What is the population of arizona?
208 Whats the largest city?
209 What is the biggest city in louisiana?
210 How many people live in austin?
211 What is the total area of the usa?
212 What is the highest point in kansas?
213 Which states border new york?
214 What state has the highest elevation?
215 What is the highest point of the state with the largest area?
216 How many people live in washington?
217 How many people live in hawaii?
218 What rivers run through new york?
219 How many people live in riverside?
220 What is the population of texas?
221 Which states border arizona?
222 What is the area of the smallest state?
223 Which state border kentucky?
224 What states border kentucky?
225 What is the largest state capital in population?
226 What is the smallest state in the usa?
227 Where is the highest point in hawaii?
228 What is the smallest city in hawaii?
229 What is the population of portland maine?
230 What are the populations of states through Which the mississippi river runs?
231 What is the shortest river?
232 What is the population of idaho?
233 What is the population of erie pennsylvania?
234 How many major rivers cross ohio?
235 What is the population of montana?
236 Which state is kalamazoo in?
237 What are the rivers in alaska?
238 Which state is the smallest?
239 What states surround kentucky?
240 Which state has the greatest population?
241 What is the area of idaho?
242 What rivers run through west virginia?
243 What is the highest point in the state with the capital des moines?

244 What length is the mississippi?
245 What is the shortest river in iowa?
246 What states border ohio?
247 What is the combined area of all 50 states?
248 What is the longest river in texas?
249 What is the population of boston massachusetts?
250 What is the capital of the state with the largest population?

Referencias

[Aguirre, 2014] Aguirre, “Modelo Semánticamente Enriquecido de Bases de Datos para su Explotación por Interfaces de Lenguaje Natural”, tesis de doctorado, División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Ciudad Madero, Cd. Madero, México, 2014.

[Anand, 2017] Probin Anand, Zuber Farooqui, “Rule based Domain Specific Semantic Analysis for Natural Language Interface for Database”, International Journal of Computer Applications (0975 – 8887) Vol. 164 – No 11, Abril 2017.

[Androutsopoulos, 1995] I. Androutsopoulos, G. Ritchie, P. Thanisch, “Natural Language Interface to Database: An Introduction”, Journal of Natural Language Engineering, pp. 29-81, 1995.

[Damljanovic, 2011] Danica D. Damljanovic, “Natural Language Interfaces to Conceptual Models”, The University of Sheffield, Julio 2011.

[Faudón, 2001] S. R. Faudón. Introducción a los Sistemas de bases de datos, 7ª edición, Pearson Educación, 2001.

[González, 2005] J. González, Traductor de Lenguaje Natural Español a SQL para un Sistema de Consultas a Bases de Datos, tesis de doctorado, Depto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México, 2005.

[González, 2018] G. González, Implementación de un Wizard para una Interfaz de Lenguaje Natural a Bases de Datos para su Consulta Mediante Internet, tesis de maestría, Depto. de Posgrado e Investigación, Instituto Tecnológico de Cd. Madero, CD. Madero, México, 2018.

[Hendrix, 1982] G. Hendrix. Natural Language Interface (panel). Computational Linguistics, 8(2):55–61, Abril–Junio 1982.

[Katz, 2018] Boris Katz, Gary Borchardt, Sue Felshin, and Federico Mora, “A Natural Language Interface for Mobile Devices”, In Kent L. Norman and Jurek Kirakowski (Eds.), The Wiley Handbook of Human Computer Interaction, Volume 2, First Edition, John Wiley & Sons, 2018, pp. 539–559.

[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.

[Mellado, 2014] O. Mellado, Implementación de un Analizador Sintáctico del Idioma Español para una Interfaz de Lenguaje Natural a Bases de Datos, tesis de maestría, Depto. de Posgrado e Investigación, Instituto Tecnológico de Cd. Madero, CD. Madero, México, 2014.

[Mittal, 2018] Ashish Mittal, Jaydeep Sen, Diptikalyan Saha, Karthik Sankaranarayanan, “An Ontology based Dialog Interface to Database”, SIGMOD’18, junio 2018.

[Osorio, 2008] F. R. Osorio, Bases de Datos Relacionales. Teoría y Práctica, 1ª edición, editorial Instituto Tecnológico Metropolitano, 2008.

[Pazos, 2013] Pazos R., González, J., Aguirre M., Martínez J. y Fraire H. “Natural Language Interfaces to Databases: An Analysis of the State of the Art”. En Proc. International Seminar on Computational Intelligence 2013. Vol. 451/2, pp. 463-480.

[RAE, 2020] Diccionario de la Real Academia Española (RAE). Recuperado de https://dle.rae.es/srv/search?m=30&w=par%C3%A1frasis (septiembre, 2020).

[Rojas, 2009] J. Rojas, Administrador de Diálogo para una Interfaz de Lenguaje Natural a Bases de Datos, disertación doctoral, Depto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 2009.

[Shabaz, 2015] K. Shabaz, Jim D. O'Shea, Keeley A. Crockett, A. Latham, “Aneesah: A Conversational Natural Language Interface to Databases”, Proceedings of the World Congress on Engineering, 2015.

[Sujatha, 2012] B.Sujatha, Viswanadha Raju, Humera Shaziya, "A Survey of Natural Language Interface to Database Management System", International Journal of Science and Advanced Technology, (ISSN 2221-8386) Volume 2 No 6, 2012.

[Tennant, 1983] H.R. Tennant, K.M. Ross, M. Saenz, C.W. Thompson, and J.R. Miller. Menu-Based Natural Language Understanding. In Proceedings of the 21st Annual Meeting of ACL, Cambridge, Massachusetts, pages 151–158, 1983.