

**UNIVERSIDAD NACIONAL DEL ALTIPLANO**

**ESCUELA DE POSGRADO**

**PROGRAMA DE MAESTRÍA**

**MAESTRÍA EN INFORMÁTICA**



**TESIS**

**APLICACIÓN DE UNA INTERFAZ DE LENGUAJE NATURAL EN EL USO  
DE LA BASE DE DATOS RELACIONAL DEL SISTEMA DE REGISTRO  
ACADÉMICO INSTITUTO DE EDUCACIÓN SUPERIOR TECNOLÓGICO  
PÚBLICO JULI, 2014**

**PRESENTADA POR:**

**RAUL ELIAS ROSADO EXELLMES**

**PARA OPTAR EL GRADO ACADÉMICO DE:**

**MAGÍSTER SCIENTIAE EN INFORMÁTICA  
MENCIÓN EN GERENCIA DE TECNOLOGÍAS DE INFORMACIÓN Y  
COMUNICACIONES**

**PUNO, PERÚ**

**2017**

**UNIVERSIDAD NACIONAL DEL ALTIPLANO**

**ESCUELA DE POSGRADO  
PROGRAMA DE MAESTRÍA  
MAESTRÍA EN INFORMÁTICA**



**TESIS**

**APLICACIÓN DE UNA INTERFAZ DE LENGUAJE NATURAL EN EL USO  
DE LA BASE DE DATOS RELACIONAL DEL SISTEMA DE REGISTRO  
ACADÉMICO INSTITUTO DE EDUCACIÓN SUPERIOR TECNOLÓGICO  
PÚBLICO JULI, 2014**

**PRESENTADA POR:**

**RAUL ELIAS ROSADO EXELLMES**

**PARA OPTAR EL GRADO ACADÉMICO DE:**

**MAGÍSTER SCIENTIAE EN INFORMÁTICA  
MENCIÓN EN GERENCIA DE TECNOLOGÍAS DE INFORMACIÓN Y  
COMUNICACIONES**

**APROBADA POR EL SIGUIENTE JURADO:**

**PRESIDENTE**

  
.....  
Dr. EDGAR CARPIO VARGAS

**PRIMER MIEMBRO**

  
.....  
D.Sc. PERCY HUATA PANCA

**SEGUNDO MIEMBRO**

  
.....  
M.Sc. SAMUEL D. PÉREZ QUISPE

**ASESOR DE TESIS**

  
.....  
M.Sc. GUIDO HUMBERTO CAYO CABRERA

Puno. 10 de Mayo de 2017

**ÁREA:** Ingeniería de software.

**TEMA:** Lenguaje y base de datos relacional.

## DEDICATORIA

*A la S. V. María, cuyos cuidados e  
inspiración son menester a lo largo de  
la vida entera.*

*A F. P. Urraca y Otros por su compañía  
desde que inició ésta tarea, que intenta  
ser más de lo que se ve a simple vista.*

*A Hugo, porque supiste inculcar el  
constante anhelo de aprender.*

*Y a Margarita, porque ningún esfuerzo  
se hace en vano.*

## AGRADECIMIENTOS

- ❖ Al asesor de esta tesis por su desinteresado esmero al brindar siempre sugerencias, tiempo y amistad.
- ❖ Al Dr. Francisco Charaja por sus aportes y destreza.
- ❖ Al presidente, primer y segundo miembros del jurado revisor de este trabajo, porque los consejos son siempre para mejorar.
- ❖ A la Escuela de Posgrado de la Universidad Nacional del Altiplano, sus docentes, administrativos y, en particular, a su Maestría en Informática.
- ❖ Y, finalmente, a ti amable lector que tienes a bien dirigir una mirada sobre este modesto trabajo.

**ÍNDICE GENERAL**

DEDICATORIA .....	i
AGRADECIMIENTOS.....	ii
ÍNDICE GENERAL.....	iii
ÍNDICE DE CUADROS .....	vii
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE ANEXOS.....	xiii
RESUMEN.....	xiv
ABSTRACT.....	xv
INTRODUCCIÓN.....	1

**CAPÍTULO I****PROBLEMÁTICA DE LA INVESTIGACIÓN**

1.1. PLANTEAMIENTO DEL PROBLEMA.....	3
1.1.1. Descripción del problema .....	3
1.2. DEFINICIÓN DEL PROBLEMA .....	5
1.2.1. Problema general .....	5
1.2.2. Problemas específicos.....	5
1.3. JUSTIFICACIÓN .....	6
1.4. OBJETIVOS DE LA INVESTIGACIÓN .....	6
1.4.1. Objetivo general .....	6
1.4.2. Objetivos específicos.....	6
1.5. HIPÓTESIS .....	7
1.5.1. Hipótesis general .....	7
1.5.2. Hipótesis específica.....	7
1.5.3. Operacionalización de variables .....	8

**CAPÍTULO II****MARCO TEÓRICO**

2.1. ANTECEDENTES .....	9
2.2. MARCO REFERENCIAL .....	10
2.2.1. Conceptos previos .....	10
2.2.2. Interfaz de lenguaje natural .....	13
2.2.3. Uso de las bases de datos relacionales .....	16
2.3. MARCO CONCEPTUAL.....	18

**CAPÍTULO III****METODOLOGÍA**

3.1. TIPO Y DISEÑO DE INVESTIGACIÓN .....	20
3.1.1. Tipo .....	20
3.1.2. Diseño .....	21
3.2. POBLACIÓN Y MUESTRA DE LA INVESTIGACIÓN .....	21
3.2.1. Población de la investigación.....	21
3.2.2. Muestra de la investigación .....	21
3.3. ÁMBITO O LUGAR DE ESTUDIO .....	22
3.4. TÉCNICAS E INSTRUMENTOS PARA RECOLECTAR INFORMACIÓN .....	22
3.4.1. Técnicas .....	22
3.4.2. Instrumentos .....	23
3.5. TÉCNICAS PARA PROCESAR Y ANALIZAR DATOS .....	23
3.5.1. Para procesar datos .....	23
3.5.2. Para analizar e interpretar datos.....	23
3.5.3. Para la presentación de datos .....	25

3.6. PROCEDIMIENTO DEL EXPERIMENTO.....	25
3.7. PLAN DE TRATAMIENTO DE DATOS.....	26
3.8. DISEÑO ESTADÍSTICO PARA LA PRUEBA DE HIPÓTESIS.....	26

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

4.1. INTERFAZ DE LENGUAJE NATURAL.....	28
4.1.1. Análisis.....	28
4.1.2. Diseño.....	47
4.1.3. Implementación.....	63
4.1.4. Test y pruebas.....	68
4.2. RESULTADOS DE LA PRUEBA DE PERFORMANCE.....	71
4.2.1. Respecto del primer enunciado de la prueba de performance.....	72
4.2.2. Respecto del segundo enunciado de la prueba de performance.....	73
4.2.3. Respecto del tercer enunciado de la prueba de performance.....	74
4.2.4. Respecto del cuarto enunciado de la prueba de performance.....	76
4.2.5. Respecto del quinto enunciado de la prueba de performance.....	77
4.2.6. Respecto del sexto enunciado de la prueba de performance.....	78
4.2.7. Respecto del séptimo enunciado de la prueba de performance.....	79
4.2.8. Respecto del octavo enunciado de la prueba de performance.....	80
4.2.9. Respecto del noveno enunciado de la prueba de performance.....	82
4.2.10. Respecto del décimo enunciado de la prueba de performance.....	83
4.3. PRUEBA DE HIPÓTESIS.....	84
CONCLUSIONES.....	86
RECOMENDACIONES.....	88
BIBLIOGRAFÍA.....	90



REFERENCIAS WEB .....	93
ANEXOS.....	95

**ÍNDICE DE CUADROS**

1. Operacionalización de variables .....	8
2. Ventajas y desventajas del uso de una ILNBD .....	15
3. Distribución de la muestra de la investigación .....	22
4. Planteamiento de los alcances de la ILN .....	28
5. Resumen de requisitos de la ILN .....	30
6. Puntuación de factores de calidad según usuarios del software .....	32
7. Cálculo de los factores de calidad .....	33
8. Detalle de los actores que interactúan con la ILN .....	35
9. Descripción del caso de uso “Iniciar sesión” .....	36
10. Descripción del caso de uso “Consultar información” .....	37
11. Descripción del caso de uso “Traducir LN a SQL” .....	38
12. Descripción del caso de uso “Consultar base de datos” .....	39
13. Descripción del caso de uso “Mostrar resultados” .....	40
14. Identificación de las clases de límite de la aplicación .....	41
15. Identificación de las clases de control de la aplicación .....	42
16. Identificación de las clases de entidad de la aplicación .....	44
17. Factores de ponderación .....	45
18. Valores de ajuste de la complejidad .....	45
19. Evaluación de horas y líneas de código por lenguaje .....	46
20. Cálculo de complejidad en el paquete Presentación.....	61
21. Cálculo de complejidad en el paquete Negocio .....	61
22. Cálculo de complejidad en el módulo Datos .....	62
23. Complejidad del software ILN .....	63
24. Pruebas de acceso a la base de datos .....	69

25. Pruebas sobre la aplicación.....	69
26. Resumen de indicadores de la métrica de prueba .....	70
27. Resultados de la prueba de performance presentada en el Anexo 4 .....	71
28. Frecuencias de las respuestas del primer enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	72
29. Frecuencias de las respuestas del segundo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	73
30. Frecuencias de las respuestas del tercer enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	74
31. Frecuencias de las respuestas del cuarto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	76
32. Frecuencias de las respuestas del quinto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	77
33. Frecuencias de las respuestas del sexto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	78
34. Frecuencias de las respuestas del séptimo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	79

35. Frecuencias de las respuestas del octavo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	80
36. Frecuencias de las respuestas del noveno enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	82
37. Frecuencias de las respuestas del décimo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	83
38. Resumen de datos para el cálculo de $Z_c$ .....	85
39. Enumeración de los vértices del grafo anterior .....	97
40. Matriz de adyacencias del grafo dirigido .....	97
41. Matriz de trazabilidad entre requisitos de usuario y requisitos de software.....	113
42. Matriz de trazabilidad entre requisitos funcionales de software y casos de uso .....	113
43. Elementos gramaticales .....	115
44. Tipos de consulta .....	125

## ÍNDICE DE FIGURAS

1. Diagrama del diseño preprueba/posprueba .....	21
2. Diagrama de actividades de la ILN .....	29
3. Diagrama de casos de uso .....	36
4. Diagrama de colaboración “Iniciar sesión” .....	47
5. Diagrama de colaboración “Consultar base de datos” .....	48
6. Diagrama de colaboración “Traducir LN a SQL” .....	48
7. Diagrama de colaboración “Consultar base de datos” .....	49
8. Diagrama de colaboración “Visualizar resultados” .....	49
9. Diagrama de secuencias principal de la ILN .....	50
10. Arquitectura de tres capas .....	52
11. Organización interna de las capas de la ILN.....	52
12. Estructura interna del paquete Presentacion .....	53
13. Estructura interna del paquete Negocio .....	53
14. Estructura interna del sub sistema NI2Sql .....	54
15. Estructura interna del paquete Datos.....	54
16. Esquema del diseño del formulario de identificación .....	55
17. Diseño de la pantalla principal de la ILN.....	56
18. Esquema de la cabecera de la pantalla principal .....	56
19. Modelo de cajas de la zona central de la pantalla principal de la ILN .....	57
20. Esquema de la pantalla principal de la ILN .....	57
21. Diagrama reticular de la ventana de identificación.....	58
22. Diagrama reticular de la pantalla principal de la ILN .....	58
25. Diseño de la base de datos de la ILN .....	59
26. Grafo direccional de búsqueda .....	59

27. Red de transiciones para la ILN.....	60
28. Captura del aspecto inicial del sistema de búsqueda .....	66
29. Captura de la pantalla principal .....	67
30. Frecuencias de las respuestas del primer enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	73
31. Frecuencias de las respuestas del segundo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	74
32. Frecuencias de las respuestas del tercer enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	75
33. Frecuencias de las respuestas del cuarto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	76
34. Frecuencias de las respuestas del quinto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	77
35. Frecuencias de las respuestas del sexto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	78
36. Frecuencias de las respuestas del séptimo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	80

37. Frecuencias de las respuestas del octavo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	81
38. Frecuencias de las respuestas del noveno enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	82
39. Frecuencias de las respuestas del décimo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016.....	83
40. Grafo dirigido.....	97

**ÍNDICE DE ANEXOS**

1. Glosario de acrónimos de informática.....	96
2. Breve reseña sobre representación y recorrido de grafos.....	97
3. Usuario informático.....	99
4. Prueba de performance .....	100
5. Catálogo de requisitos .....	103
6. Validación de software por parte de los usuarios.....	114
7. Clasificación de los elementos gramaticales.....	115
8. Código fuente de algunos módulos de la ILN .....	116
9. Breve manual del usuario de la ILN .....	120
10. Clasificación del tipo de consulta .....	125
11. Matriz de consistencia .....	126

## RESUMEN

Este documento trata sobre la elaboración e implementación de un software Interfaz de Lenguaje Natural para Base de Datos que traduce consultas textuales del usuario, hechas mediante el lenguaje común que es empleado por las personas en sus comunicaciones diarias (el castellano), a un equivalente semántico en un lenguaje aplicado y entendido por los ordenadores (el SQL); con la finalidad de poner en el mismo nivel procedimental a todos los usuarios que tienen que trabajar con la información almacenada en una base de datos. Para tal efecto, se tiene como objetivo primordial determinar la medida en que dicha interfaz mejora el uso de las bases de datos relacionales. Los resultados obtenidos muestran un modo alternativo y más sencillo de buscar y recuperar información sin entendimiento ni capacitación al respecto. De modo que el usuario pueda hacerlo sin aprender a programar. Esta investigación llegó a concluir que el software “Interfaz de Lenguaje Natural” optimiza el uso de las bases de datos. Los beneficios de este estudio radican en la novedad del tema dentro del ámbito investigativo local, y la diversidad de áreas en las cuales también se podría aprovechar la interfaz.

**Palabras clave:** Búsqueda, información, interfaz de lenguaje natural, lenguaje, recuperación, uso de bases de datos, usuario.

## ABSTRACT

This document deals with the elaboration and implementation of a software Natural Language Interface to Database that translates the user's textual consults, made by means of the ordinary language that is used by people in its daily communications (the Spanish), to a language semantically equivalent applied and understood by the computers (SQL); with the purpose of putting in the same procedural level to all the user's that have to work with the information stored in a database. For such an effect, one has as primordial objective to determine the measure in that this interface improves the use of the relational databases. The obtained results show an alternative and simpler way to search and retrieve understanding and training in this respect. He or she can make it without learning how to program. This investigation reached the conclusion that the software "Interface of Natural Language" optimizes the use of the databases. The benefits of this study reside in the novelty of the topic inside the local investigative environment, and the diversity of areas in which it could also take advantage the interface.

**Keywords:** Database use, information, language, natural language interface, retrieve, search, user.

## INTRODUCCIÓN

Este documento presenta el desarrollo de una de las sugerencias expuestas en una investigación previa, que habla sobre la posibilidad de aplicar el Procesamiento del Lenguaje Natural en la enseñanza.

Hablando más específicamente del entorno investigativo local, es sabido que el nivel de capacitación obtenido sobre los conceptos relacionados a las bases de datos, y la disponibilidad de gente bien preparada sobre esta materia no se equilibran. Tal realidad justificó emplear mecanismos que, facilitando el uso computacional, optimizaran la enseñanza de las bases de datos, aunque tan sólo fuera de forma experimental; aplicándolos en una institución educativa que brindara las condiciones adecuadas para ello.

El resultado que dejó dicha investigación se documenta en las siguientes páginas, las cuales terminan por sostener que: aprovechar un software procesador del lenguaje humano mejora la manipulación de las bases de datos; resultando especialmente apropiado en personas con pocos conocimientos sobre programación de bases de datos.

Pero, este documento va más allá de la simple prueba, describe también la adaptación e implementación, a grandes rasgos, de dicho software; para aportar una mirada más “íntima” sobre este tipo de programas computacionales. Dándole especial relevancia a las métricas de software, para dejar constancia también de su calidad.

Por todo ello, y para mejor entendimiento, el presente trabajo se organiza así:

El Capítulo I describe en profundidad la problemática, objetivos, motivaciones e hipótesis.

El Capítulo II presenta el marco teórico, así como los trabajos relacionados y se definen los términos de mayor significado.

El Capítulo III expone el tipo y diseño de investigación, el procedimiento del experimento, población, muestra, ámbito; las técnicas, instrumentos y material requerido para recabar datos y procesarlos; así como el diseño estadístico adoptado con el fin de probar la hipótesis.

En el Capítulo IV se describen las distintas fases de la construcción del software, como elemento por antonomasia para recabar datos y entender su efecto sobre la muestra. Y, posteriormente, analizar los datos y establecer su correspondiente descripción e interpretación.

Las Conclusiones sintetizan los resultados sobre el desarrollo, uso y experimentación con la interfaz de lenguaje natural para contrastar la hipótesis.

Las Recomendaciones plantean líneas investigativas para futuros estudios sobre el mismo campo, y exponen algunos consejos para mejorar el software.

Por último, los Anexos exhiben información que sobrepasa el contenido mismo de esta obra, pero que puede ser consultada para más detalle.

## CAPÍTULO I

### PROBLEMÁTICA DE INVESTIGACIÓN

#### 1.1. PLANTEAMIENTO DEL PROBLEMA

##### 1.1.1. Descripción del problema

Desde su apertura en 1982, el Instituto de Educación Superior Tecnológico Público Juli (IESTP Juli) tuvo entre sus fines la formación de técnicos profesionales para crear, adecuar y utilizar tecnologías que atiendan las exigencias del desarrollo social; destacando para ello, de entre las carreras que oferta, la de Computación e Informática (CI).

En el último semestre de formación el Instituto exige realizar prácticas pre-profesionales a sus alumnos, de modo que esta experiencia les sirva para empezar su desenvolvimiento profesional.

Por ello, y como particular aporte a sus estudiantes, dicho Instituto realiza concursos anuales para cubrir plazas de practicantes, según se necesite, en áreas afines a la informática, particularmente en el Sistema de Registro Académico que es el que más asistencia requiere por la

necesidad de interactuar con su base de datos, merced a la gran cantidad de reportes que se deben de elaborar.

Pero, una vez escogidos y puestos a realizar sus respectivos deberes, los ya practicantes, manifestaban una desigual preparación en el manejo de bases de datos, situación que obligaba la asidua supervisión y apoyo de gente más experimentada (p.ej. un docente), sin cuyo concurso los errores involuntarios crecían cerca de un 30%, según un estudio sobre la calidad de atención al público usuario hecha un año antes.

Para mitigar la carencia de buenos colaboradores se empezó a extender la convocatoria de prácticas a egresados de otras instituciones educativas, hecho inaceptable para una institución que debería preciarse de adiestrar técnicos idóneos en el área de la informática.

De haber persistido esta situación, se imposibilitaría la acreditación de la entidad, ya que la falta de dominio de los estudiantes sobre ciertos temas clave hubiera degenerado en una dependencia crónica de gente más conocedora, lo que motivaría la creencia de que el Instituto era incapaz de forjar buenos profesionales.

Aunque los estudiantes de Computación e Informática están obligados a conocer tecnologías como las bases de datos, ya que son cruciales en áreas con sistemas computarizados, para crear reportes se requiere enunciar complejas instrucciones de ordenador que complican a quienes carecen de buen entendimiento sobre programación de computadoras.

De modo que el problema residía en crear algún mecanismo que permitiera consultar bases de datos de forma “más humana” (como si de

un pseudocódigo se tratara), para aprender éste tema. Ello significaba desarrollar, aprovechando una metodología óptima, un software procesador del lenguaje humano. Tal razonamiento condujo a formular las siguientes preguntas para ser investigadas:

## **1.2. DEFINICIÓN DEL PROBLEMA**

### **1.2.1. Problema general**

¿En qué medida mejoran en el uso de las bases de datos relacionales los usuarios del Sistema de Registro Académico del Instituto de Educación Superior Tecnológico Público Juli con la aplicación de una interfaz de lenguaje natural?

### **1.2.2. Problemas específicos**

- ¿Cuál es el nivel que alcanza el análisis orientado a objetos respecto de una interfaz de lenguaje natural para el uso de una base de datos relacional?
- ¿Cuál es el nivel que logra el diseño orientado a objetos respecto de una interfaz de lenguaje natural para usarla sobre una base de datos relacional?
- ¿En qué medida se implementa mediante programación orientada a objetos y se prueba una interfaz de lenguaje natural sobre una base de datos relacional?
- ¿Qué nivel de manejo procedimental muestran los estudiantes del Instituto de Educación Superior Tecnológico Público Juli respecto del

uso de una base de datos relacional antes y después de experimentar con una interfaz de lenguaje natural?

### **1.3. JUSTIFICACIÓN**

Es importante facilitar el uso procedimental de las bases de datos, por ser un tema que exige técnica para saber recabar de ellas la información qué y cómo se desea. Una interfaz de lenguaje natural ayuda a sortear dicha exigencia ya que, a partir de sus propias expresiones (y no de las que pide la computadora), el usuario se familiariza mejor con la elaboración correcta de consultas SQL. Debido a eso, este trabajo adquiere relevancia metódica, ya que aporta una forma alternativa de enseñanza asentada en el PLN.

### **1.4. OBJETIVOS DE LA INVESTIGACIÓN**

#### **1.4.1. Objetivo general**

Determinar la medida en que mejoran en el uso de bases de datos relacionales los usuarios del Sistema de Registro Académico del Instituto de Educación Superior Tecnológico Público Juli con la aplicación de una interfaz de lenguaje natural.

#### **1.4.2. Objetivos específicos**

- Determinar el nivel que alcanza el análisis orientado a objetos respecto de una interfaz de lenguaje natural para el uso de una base de datos relacional.
- Evaluar el nivel logrado por el diseño orientado a objetos respecto de una interfaz de lenguaje natural para usarla sobre una base de datos relacional.

- Señalar la medida en que se implementa mediante programación orientada a objetos y se prueba una interfaz de lenguaje natural sobre una base de datos relacional.
- Evaluar el nivel de manejo procedimental que muestran los estudiantes del Instituto de Educación Superior Tecnológico Público Juli respecto del uso de una base de datos relacional antes y después de experimentar con una interfaz de lenguaje natural.

## 1.5. HIPÓTESIS

### 1.5.1. Hipótesis general

La mayoría de los usuarios del Sistema de Registro Académico del Instituto de Educación Superior Tecnológico Público Juli mejoran significativamente en el uso de las bases de datos relacionales con la aplicación de una interfaz de lenguaje natural.

### 1.5.2. Hipótesis específica

- El análisis orientado a objetos respecto de una interfaz de lenguaje natural para el uso de una base de datos relacional alcanza un nivel aceptable.
- El diseño orientado a objetos respecto de una interfaz de lenguaje natural para usarla sobre una base de datos relacional logra un nivel aceptable.

- La implementación mediante programación orientada a objetos y prueba de una interfaz de lenguaje natural sobre una base de datos relacional corresponden a un nivel aceptable.
- El nivel de manejo procedimental que muestran los estudiantes del Instituto de Educación Superior Tecnológico Público Juli respecto del uso de las bases de datos relacionales antes del experimento es del nivel Deficiente; mientras que después del mismo es del nivel Bueno.

### 1.5.3. Operacionalización de variables

**Cuadro 1.** Operacionalización de variables

Variable	Dimensión	Indicador	Categoría	Índice	Instrumento
V.I. Interfaz de lenguaje natural	Desarrollo de software orientado a objetos	Análisis, diseño, implementación y prueba	<ul style="list-style-type: none"> <li>• Aceptable</li> <li>• No aceptable</li> </ul>	Métricas de software	Métrica de McCall, Puntos de función, Métrica de diseño y Métrica de Halstead
V.D. Uso de las bases de datos relacionales	Procedimental	Manipulación	<ul style="list-style-type: none"> <li>• Muy deficiente 00- 05</li> <li>• Deficiente 06-10</li> <li>• Regular 11-13</li> <li>• Bueno 14-17</li> <li>• Muy bueno 18-20</li> </ul>	Nivel de aprendizaje	Prueba de performance

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1. ANTECEDENTES

Muchos y muy variados han sido los acercamientos al PLN, tópico surgido en la década de los años cuarenta del siglo XX, de entre ellos, por su cercanía cronológica y destinarse al ámbito hispano, se menciona a Pereiras (2011), quien desarrolló un sistema de recuperación de información con técnicas de procesamiento de lenguaje, para convertir pedidos en español a un lenguaje de consultas sobre información relativa a las historias clínicas de autopsias en un hospital. Dicho sistema, aceptaba variedad de consultas y ofrecía resultados adecuados incorporando una herramienta que verificaba la sintaxis de las preguntas hechas.

También, por el mismo tiempo Cócera (2011) desarrolla un software integrado al MS Office para interactuar como un componente multilenguaje que realiza operaciones mediante lenguaje natural (castellano o inglés).

En 2012 Zamoszczyk, De Luca, Ruiz e Iturbide presentan el desarrollo de una herramienta que intenta mejorar la comunicación hombre-máquina,

valiéndose de una interfaz para web y *smartphones*, capaz de generar SQL interpretando consultas hechas por medio del lenguaje natural escrito u oral; de manera que muestre los resultados en forma de texto, lista o gráfico, y se integren a cualquier software de gestión de negocios.

Posteriormente, Alconada (2013) desarrolla otra herramienta capaz de convertir el lenguaje natural en sentencias SQL para ser procesadas después por el gestor de bases de datos MySQL.

Si bien todos los estudios citados proceden del ámbito extranjero, estos pueden encontrar su semejante en el ámbito local. A la fecha, Rosado (2016) desarrolló una interfaz de lenguaje para buscar datos en un sistema de información previamente implementado.

## **2.2. MARCO REFERENCIAL**

### **2.2.1. Conceptos previos**

#### **2.2.1.1. Lenguaje natural vs lenguaje formal**

Russell y Norvig (2010) dicen respecto del Lenguaje Natural, LN, también llamado Lenguaje Humano, LH, (véanse más acrónimos en el Anexo 1), que es usado de continuo por una comunidad de hablantes para comunicarse entre sí (como el castellano, chino o inglés), no teniendo definiciones estrictas. Por el contrario, el Lenguaje Formal, LF, sí las tiene (p.ej. en el lenguaje Python 3 la sentencia “`print(2+2)`” es correcta; mientras que “`)2+2(print`”, no).

### **2.2.1.2. Procesamiento del lenguaje natural**

Gelbukh (2010) entiende el Procesamiento de Lenguaje Natural (PLN) como la habilidad de una máquina para procesar la información comunicada, y no únicamente los caracteres o sonidos. Así, si un texto se transmite a un buscador de documentos, y éste devuelve sólo los que contienen dicho texto no sería un software PLN. Pero, si entiende el significado subyacente tanto en el texto como de cada documento, y escoge aquellos que tienen las mismas ideas, entonces, sí sería una aplicación PLN.

Moreno (2009), por su parte, especifica que el PLN representa el desarrollo de modelos computacionales de determinados aspectos del lenguaje humano (ya que el PLN carece de modelos completos que entiendan el habla humana). A partir de estos modelos se pretenden realizar programas informáticos capaces de comprender o producir enunciados en lenguaje natural.

### **2.2.1.3. Arquitectura de un sistema PLN**

J. J. González, Morales y Florencia (2012) señalan que la arquitectura PLN se basa en una definición por fases. Dichas fases (también nombradas “módulos”, “procesos”, “análisis”, “etapas” o “conocimientos”, por otros autores) son: 1] Morfológico, que determina la función de cada palabra en la oración. 2] Sintáctico, que analiza la correcta combinación de palabras. 3] Semántico, que da los posibles significados de la oración. 4] Contextual (o

pragmático, o del discurso), que discierne entre los significados y escoge el más apropiado de acuerdo al contexto.

#### **2.2.1.4. Gramáticas regulares o de estados finitos**

Autores como Barón y Müller (2014) explican que Chomsky concibió la revolucionaria idea de entender al lenguaje como un conjunto infinito de oraciones generadas por reglas de buena formación: la gramática. Esta, acotan Russell y Norvig (2010), puede percibirse como una Red de Transiciones creadora de todas las oraciones de una sola lengua.

Una red de transiciones es algo así como un mapa con distintos caminos que permite encontrar las expresiones gramaticales de una lengua. La red está formada por una serie de nodos o estados (entre los que hay uno inicial, y uno o varios finales), unidos por arcos (los caminos o transiciones entre estados) etiquetados con símbolos terminales (palabras o categorías léxicas); de manera que una oración pertenecerá a la lengua definida por la red si hay un camino desde el estado inicial hasta uno final.

#### **2.2.1.5. Tipo de dato abstracto grafo**

Goodrich, Tamassia y Goldswasser (2014) definen un grafo “G” como una serie de conexiones (arcos o aristas) “E”, en un conjunto “V” de pares de objetos (vértices o nodos). Las aristas pueden ser: dirigidas, si los elementos de la arista (u, v) en V están ordenados y “u” precede a “v”; o no dirigidas, si aquellos elementos no están ordenados, en éste caso (u, v) se trata igual a (v, u). Los grafos

suelen visualizarse trazando los vértices como óvalos o rectángulos, y las aristas como líneas que unen dos vértices.

Un grafo se representa de muchas formas. Una que destaca por su sencillez es la Matriz de Adyacencias. Asimismo, para recorrer un grafo (conocer sus conexiones) existen varias técnicas, pero para esta tesis bastará la “Primera búsqueda en profundidad”, que sirve para grafos no dirigidos. Como esta investigación no pretende ahondar en tal tema se recomienda leer el Anexo 2.

#### **2.2.1.6. Métricas de software**

Una métrica es cualquier medida o conjunto de ellas destinada a conocer o estimar el tamaño u otra característica de un software, generalmente para realizar comparativas o planificar proyectos de desarrollo (Pressman, 2010).

#### **2.2.2. Interfaz de lenguaje natural**

Conforme a Rodríguez (2003) una Interfaz de Lenguaje Natural, ILN, (también conocida como Interfaz de Usuario de Lenguaje Natural, Interfaz de Lenguaje Humano, y otros) es un tipo de interfaz (mecanismo de comunicación entre un usuario y una aplicación informática) que usa la lengua como elemento básico, no necesariamente único, de comunicación. Es decir, este tipo de interfaz controla una aplicación software, principalmente, por medio del lenguaje natural.

López-Cózar (2013) agrega que esta clase de sistemas pretenden emular la inteligencia humana, de modo que tendrán predisposición para

labores bien definidas que parecen destinadas sólo a las capacidades humanas: como el entendimiento del lenguaje común.

Para J. J. González, et ál. (2012) Una ILN intenta simplificar la interacción hombre-máquina, por tanto, están pensadas para que usuarios sin mayores conocimientos técnicos puedan interactuar con un sistema computacional por medio del lenguaje.

Cócera (2011), observa que, en usuarios ocasionales, el uso del lenguaje natural brinda formas de comunicación efectiva (p.ej., acceder a datos). Pero dichos usuarios deben tener conocimientos exactos de la cobertura de la aplicación y al elaborar consultas (u otras operaciones) hacerlas sobre dominios restringidos.

Sobre esto último, afirman Aguirre y Pazos (2012) que las bases de datos modelan la información del mundo real y, por tanto, pueden considerarse dominios restringidos ya que según Montalbán y Castro (2014) dicha información es propia de un mismo contexto o problema.

En otro escrito, los citados, Aguirre y Pazos (2015) hablan del rol fundamental que tiene en la vida de las personas la información; siendo las mayores fuentes de esta las bases de datos, por lo que acceder a su contenido le es deseable a cualquier persona.

Dicho acceso se consigue por medio del lenguaje, aunque no el humano, ya que las computadoras no lo entienden; sino uno formal (o artificial) llamado SQL. Pero que, infortunadamente, está restringido mayoritariamente a profesionales de la computación.

Para remediar esa situación, surge un tipo especial de interfaz de lenguaje natural dedicada a consultar bases de datos (Shabaz, O'Shea, Crockett, & Latham, 2015). Estas les permiten a los usuarios buscar y recuperar información a través de una consulta en lenguaje humano; pero no a modificarla (al menos en principio), ya que, según Cócera (2011), debe regularse la integridad de la base de datos. Este tipo de interfaz, como todo software, tiene ventajas y desventajas para su uso:

**Cuadro 2.** Ventajas y desventajas del uso de una ILNBD

<b>Ventajas</b>	<b>Desventajas</b>
- Se puede expresar una consulta igual que como se la piensa	- La cobertura lingüística no es obvia
- No requiere del aprendizaje de un nuevo lenguaje	- Pueden llegar a ser un medio de comunicación difícil
- El lenguaje natural es más cómodo para el usuario	- Su configuración puede llegar a ser lenta y tediosa
- Disminuyen notablemente el tiempo de aprendizaje del usuario	- A veces el usuario supone habilidades que la interfaz no posee

Fuente: Basado en Cócera (2011)

### 2.2.2.1. Análisis orientado a objetos

El análisis orientado a objetos o AOO, enfatiza la construcción de modelos del mundo real, utilizando una visión orientada a objetos, o sea, es un método de análisis que examina los requisitos desde la perspectiva de las clases y objetos que se encuentran en el vocabulario del dominio del problema. Los productos de este

análisis sirven como modelos de los que puede partirse para hacer un diseño orientado a objetos (Booch, et ál., 2007).

#### **2.2.2.2. Diseño orientado a objetos**

Denotado como DOO, hace abstracciones de clases y objetos para estructurar lógicamente los sistemas. Este método abarca el proceso de descomposición orientada a objetos y una notación para describir los modelos lógico (estructura de clases y objetos) y físico (arquitectura de módulos y procesos), así como los modelos estático y dinámico del sistema que se diseña. Los productos del DOO pueden utilizarse como anteproyecto para la implementación orientada a objetos (Booch, et ál., 2007).

#### **2.2.2.3. Implementación o programación orientada a objetos**

Conocida abreviadamente como POO, es un método en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representa la instancia de una clase, y estas son, todas ellas, miembros de una jerarquía que se une mediante relaciones de herencia (Booch, et ál., 2007).

#### **2.2.3. Uso de las bases de datos relacionales**

Un usuario de software puede ser quien lo opera o quien está bajo su influencia (ver el Anexo 3). Cualquiera puede emplear algún programa especial en una tarea determinada no requiriendo ser un especialista. P.ej., en labores sujetas a la clasificación y hallazgo de dependencias de información, se usa un tipo de software llamado base de datos.

Las Bases de Datos (BD) son un conjunto de datos almacenados en la memoria externa y organizados mediante estructuras de datos (Marqués, 2011). Recogen información de algún objeto o suceso, y la relacionan formando vínculos que la mantienen ordenada y coherente (Montalbán y Castro, 2014). Aprovechadas inicialmente por gente entendida en procesar información, gracias a la asequibilidad de las computadoras, se han generalizado en pequeñas y medianas empresas como apunta Ricardo (2009).

Aunque hay diversos tipos de bases de datos, destacan las Relacionales. Introducidas por Codd (Marqués, 2011), su popularidad se debe a su sencillez para relacionar datos (la palabra “relacional” proviene del término matemático “relación”, equivalente a “tabla”, conformada por “filas” o registros y “columnas” o campos que cada registro tiene). Consecuentemente, cada dato se identifica inequívocamente por la intersección de la fila y la columna a la cual pertenezca; y así, los no programadores, puedan usar cómodamente grandes cantidades de datos.

Este enfoque *light* de la información contrasta con una visión más formal, la cual, señalan Camps, et ál. (2010), implica Dominio, Esquemas de relación, etc.; y que, como añaden Bender, et ál. (2014), logra incumbrir en otras áreas relacionadas con la recuperación de información como la Inteligencia Artificial (IA) y el PLN.

Usar bases de datos puede considerarse equivalente al proceso compuesto de recuperar y presentar información (Kendall & Kendall,

2011). Para esto se suele aplicar el SQL que es una herramienta con la que se define y manipula la información (Silberschatz, Korth, & Sudarshan, 2011).

### **2.2.3.1. Nivel procedimental del uso del software**

Emplear un software puede pensarse como una forma de aprendizaje, considerando elementos instruccionales orientados a contenidos específicos (el procedimental) que desarrollan en el usuario competencias definidas en la teoría constructivista adoptada por el Ministerio de Educación.

A diferencia de los conocimientos logrados leyendo o escuchando, los procedimientos se aprenden por ejercitación, pero yendo más allá del mecanicismo, es decir, un aprendizaje meramente repetitivo. Logrando que el aprendiz reflexione cada acción, de modo que su conocimiento pueda ser utilizado en otras situaciones (Sánchez, s.f.). Este aprendizaje resulta, pues, de vital importancia a la hora de valorar el uso correcto de un software.

## **2.3. MARCO CONCEPTUAL**

- Interfaz hombre-máquina. Medio que permite intercambiar información entre humanos y aparatos tecnológicos, como las computadoras. Se les diseña pensando en las necesidades humanas y dan origen a una nueva disciplina: la Interacción Persona-Ordenador, IPO, que es un conjunto de conceptos encargados de diseñar, implementar y evaluar las interfaces hombre-máquina.

- Lenguaje. Capacidad humana que utiliza un sistema expresivo oral o escrito, para comunicar ideas o sentimientos. Formalmente se le puede definir como un compuesto de todas las palabras que lo constituyen, de todas las oraciones construidas correctamente, y del significado dado a esas oraciones.
- Orientación a objetos. Visión que usa abstracciones que identifican elementos fundamentales de la realidad a los cuales denomina objetos. Estos colaboran entre sí en la construcción de una solución.
- Procedimental. Modo de englobar ciertas acciones que suelen realizarse de una misma forma, con una serie común de pasos claramente definidos, que permiten realizar una ocupación, trabajo o estudio.
- Uso (informático). Conjunto de acciones que facilitan servirse de un producto computacional para el logro de un fin.

## CAPÍTULO III

### METODOLOGÍA

#### 3.1. TIPO Y DISEÑO DE INVESTIGACIÓN

##### 3.1.1. Tipo

Por el propósito del estudio, el tipo de investigación es aplicado. Hernández, Fernández y Lucio (2010) declaran al respecto que el objeto de esta clase de investigaciones es probar la efectividad de técnicas, instrumentos o máquinas que puedan ser utilizados inmediatamente para resolver problemas o emprender proyectos empresariales productivos. Como en este estudio, donde se pretende probar la efectividad de una interfaz del lenguaje natural, con el fin de mejorar el uso de las bases de datos relacionales.

Por el criterio estratégico de investigación, el tipo al que corresponde es el experimental. Para los antedichos Hernández et ál. (2010), estas investigaciones se caracterizan por la preparación de condiciones de estudio y, sobre todo, por manipular la variable dependiente (Véase la Operacionalización de variables).

### 3.1.2. Diseño

Corresponde un diseño de un solo grupo experimental (la muestra), al que se aplica una prueba antes del tratamiento experimental para observar cómo estaba al principio. Luego, se le administra el experimento y se observa el cambio producido en él mediante una prueba final (ver la siguiente figura). El lapso entre pruebas es de tres (03) semanas, para prevenir cambios en el grupo:

G                      O<sub>1</sub>                      X                      O<sub>2</sub>

**Figura 1.** Diagrama del diseño preprueba/posprueba

Fuente: Hernández et ál. (2010, pág. 136)

Dónde,

G : Grupo de prueba

O<sub>i</sub> : Prueba

X : Experimento

## 3.2. POBLACIÓN Y MUESTRA DE LA INVESTIGACIÓN

### 3.2.1. Población de la investigación

Está conformada por los ciento cincuenta (150) estudiantes de la carrera de Computación e Informática del IESTP Juli.

### 3.2.2. Muestra de la investigación

#### 2.3.1.1. Descripción

Está constituida por los alumnos del curso de Base de Datos que pertenecen al quinto semestre de la carrera de Computación e

Informática del IESTP Juli, y que hacen uso de la interfaz de lenguaje natural. Esta población se enumera así:

**Cuadro 3.** Distribución de la muestra **de la investigación**

<b>Grupos</b>	<b>Total</b>
Varones	18
Mujeres	13
Total	31

Fuente: Nómina de estudiantes del V semestre de la carrera de Computación e Informática del IESTP Juli

#### **2.3.1.2. Método**

El método para la obtención de la muestra es de tipo no probabilístico, ya que no se escogieron a los miembros mediante ningún tipo de sorteo.

### **3.3. ÁMBITO O LUGAR DE ESTUDIO**

El ámbito espacial para el trabajo de campo corresponde al Instituto de Educación Superior Tecnológica Público Juli; ubicado en la Av. Alfonso Ugarte 1064 de la ciudad de Juli, capital de la provincia de Chucuito, Región Puno - Perú.

### **3.4. TÉCNICAS E INSTRUMENTOS PARA RECOLECTAR INFORMACIÓN**

#### **3.4.1. Técnicas**

Se usa como técnica un examen práctico, el cual formula tareas por escrito para que los miembros de la muestra las resuelvan antes y después de aplicar el experimento, con el fin de determinar el nivel de dominio procedimental que tienen respecto de las bases de datos.

### 3.4.2. Instrumentos

Corresponde aplicar una prueba de performance, ya que permite conocer y mitigar los riesgos relacionados, entre otras cosas, con el mal desempeño en el uso de las aplicaciones (véase el Anexo 4).

## 3.5. TÉCNICAS PARA PROCESAR Y ANALIZAR DATOS

### 3.5.1. Para procesar datos

- 1) Se revisan y consistencian cada uno de los datos, para prevenir descuidos o errores aparecidos al aplicar el instrumento de recolección de información.
- 2) Se clasifica la información mediante distribuciones de frecuencias.
- 3) Se codifican y tabulan, de acuerdo a la Operacionalización de variables, para observar mejor sus valores.

### 3.5.2. Para analizar e interpretar datos

En este estudio se utilizan las siguientes pruebas estadísticas.

#### 3.5.2.1. Media aritmética

Para saber cuál es el nivel promedio en la manipulación del software de los usuarios antes y después del experimento. Levine, Krehbiel y Berenson (2012) presentan la fórmula:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

Dónde:  $X_i$ , es el i-ésimo valor de  $X$ ; y  $n$ , es el tamaño de muestra.

### 3.5.2.2. Desviación estándar

Ayuda a determinar la Diferencia de medias. Siguiendo a Johnson y Kuby (2012, pág. 86), la fórmula es:

$$S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}$$

Donde  $S^2$  es la desviación estándar.  $\bar{X}$  es el promedio.  $X_i$ , el  $i$ -ésimo valor de  $X$ . Y  $n$ , el tamaño de la muestra.

### 3.5.2.3. Diferencia de medias

Para comprobar la veracidad de la hipótesis. Mediante la teoría estadística y la información disponible, se aplica el estadígrafo estimador  $Z_c$  (por ser la muestra mayor a 30):

$$Z_c = \frac{|\bar{X}_A - \bar{X}_B|}{\sqrt{\frac{S^2_A}{n_A} + \frac{S^2_B}{n_B}}}$$

Donde:

$\bar{X}_A$  : es el promedio pre test

$\bar{X}_B$  : el promedio post test

$S^2_A$  : la varianza pre test

$S^2_B$  : la varianza post test

$n_A$  : el tamaño de la muestra pre test

$n_B$  : el tamaño de la muestra post test

### 3.5.3. Para la presentación de datos

- 1) Cuadros estadísticos: ya que despliegan ordenada y sistemáticamente los promedios anteriores y posteriores al experimento.
- 2) Gráficos cónicos 3D: para comparar valores entre las calificaciones de hombres y mujeres, antes y después de aplicar el experimento. Estos tienen, además, la propiedad de dejar visualizar resultados de varias dimensiones mejor que los clásicos gráficos de barras. Se elaboran en base a los correspondientes cuadros estadísticos.

### 3.6. PROCEDIMIENTO DEL EXPERIMENTO

Primero. Se solicita el permiso de las autoridades del IESTP Juli para realizar la investigación, poniéndolos al corriente del experimento.

Segundo. Se coordina con los participantes del experimento (los miembros de la muestra) su colaboración para las dos etapas del experimento.

Tercero. Se prepara el material experimental descrito en 3.4.2, y se procede con la primera etapa del experimento, conforme el diseño del punto 3.1.2.

Cuarto. A los participantes se les aplica la prueba de performance (descrita en el Anexo 4), pidiéndoles que la resuelvan completamente mediante el uso directo y exclusivo del SQL.

Quinto. Se procede a calificar el total de puntos por estudiante, y se aplica el Plan de tratamiento de datos (visto más adelante en 3.7).

Sexto. Llegado el plazo, se aplica la segunda etapa del experimento, para lo cual en cada computadora del laboratorio de cómputo del IESTP Juli, se instala la interfaz de lenguaje natural.

Séptimo. Se repite el paso Cuarto, pero esta vez se pide a los participantes resolver la prueba de performance únicamente mediante la interfaz de lenguaje natural.

Octavo. Para los nuevos datos obtenidos se repite el paso Quinto.

Noveno. Con los datos de las dos etapas experimentales procesados se prueba la hipótesis como indica el apartado 3.8 (más adelante).

### **3.7. PLAN DE TRATAMIENTO DE DATOS**

Primero. Se procesan los datos de acuerdo al apartado 3.5.1.

Segundo. Se presentan los datos como indica el punto 3.5.3.

Tercero. Se calcula la media aritmética en base al apartado 3.5.2.1.

Cuarto. Se calcula la varianza de acuerdo al acápite 3.5.2.2.

### **3.8. DISEÑO ESTADÍSTICO PARA LA PRUEBA DE HIPÓTESIS**

En este punto se aplica la Diferencia de Medias de la siguiente forma:

Primero. Se plantean las hipótesis nula ( $H_0$ ) y alterna ( $H_1$ ).

Segundo. Se fija el nivel de significancia  $\alpha$

Tercero. Se plantea la regla de decisión.

Cuarto. Se calcula  $Z_c$  según la fórmula del apartado 3.5.2.3.

Quinto. Se aplica la regla de decisión previamente planteada.

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1. INTERFAZ DE LENGUAJE NATURAL

##### 4.1.1. Análisis

##### 4.1.1.1. Definición del software

Desarrollar un software Interfaz (en lo sucesivo sólo ILN) que use el lenguaje natural como medio para recuperar información desde una base de datos, y probarlo como instrumento educativo en la enseñanza del curso correspondiente de la Carrera de Informática del IESTP Juli, con el fin de adiestrar a los estudiantes del curso correspondiente en labores como las del Sistema de Registro Académico.

##### 4.1.1.2. Determinación del alcance del software

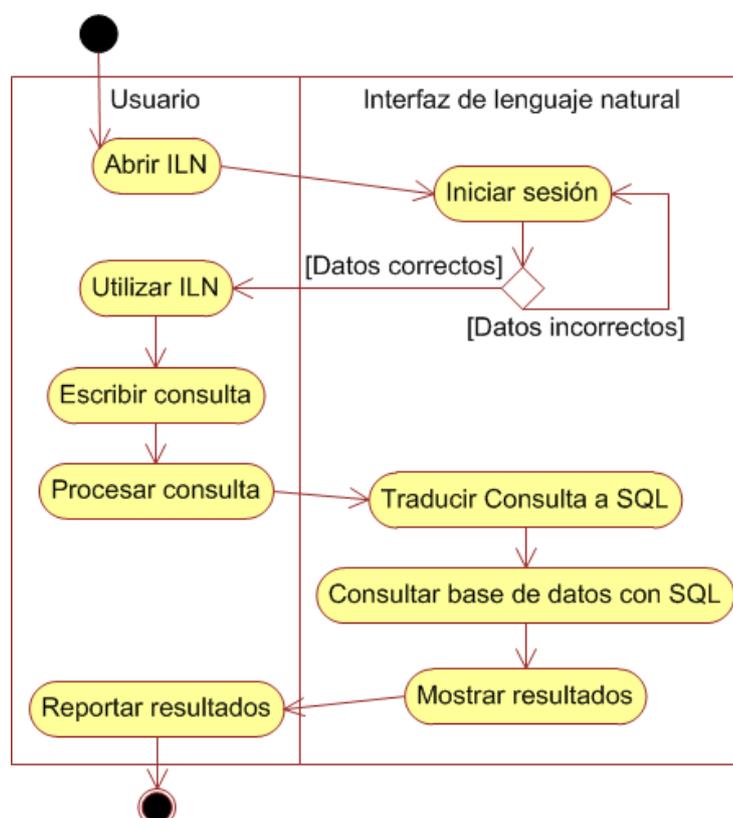
**Cuadro 4.** Planteamiento de los alcances de la ILN

Concepto	Descripción
Geográfico	- Uso exclusivo en el IESTP Juli
Tecnológico	- Aplicación del PHP y MySQL en la IA

	- Demuestra las ventajas de paradigma Orientado a Objetos
Temporal	- Terminación e implantación relativamente breves
Operativo	- Utiliza el habla castellana - Dedicada a recuperar información - Aprendizaje y aplicación más simple que el SQL

### 4.1.1.3. Modelo del proceso de negocio

Al iniciar la creación de un software debe entrevistarse a sus posibles usuarios para comprender lo que debe hacer. En este caso: cómo el usuario consulta una base de datos aprovechando la ILN. Este proceso se puede representar mediante UML.



**Figura 2.** Diagrama de actividades de la ILN

Fuente: Entrevistas a los usuarios directos

#### 4.1.1.4. Identificación de requisitos

Aprovechando las entrevistas, también se debe conocer lo que se espera del software a desarrollar, es decir, sus requisitos. Estos se clasifican en un catálogo (ver Anexo 5) que se resume en el cuadro que continúa.

**Cuadro 5.** Resumen de requisitos de la ILN

Categoría	Identificador	Nombre
Requisitos funcionales de usuario	RU-001	Acceder a la ILN
	RU-002	Escribir consulta de recuperación usando lenguaje natural
	RU-003	Escribir consulta de recuperación usando términos clave
	RU-004	Escribir consulta de recuperación usando lenguaje natural y términos clave mezclados
	RU-005	Visualizar la <i>query</i> SQL resultante
	RU-006	Visualizar datos recuperados
	RU-007	Visualizar mensajes de advertencia
	RU-008	Visualizar mensajes de error
Requisitos no funcionales de usuario	RU-009	Propósito netamente formativo
	RU-010	Facilidad de uso
	RU-011	Rapidez de aprendizaje
	RU-012	Ocultar procesamiento de la consulta del usuario
Requisitos funcionales de software	RS-001	Iniciar sesión de trabajo
	RS-002	Leer consulta escrita por el usuario
	RS-003	Esperar consulta correcta de parte del usuario
	RS-004	Permitir sólo recuperar datos

	RS-005	Formatear la consulta introducida por el usuario
	RS-006	Traducir consulta del usuario a una query SQL
	RS-007	Combinar tablas automáticamente
	RS-008	Formatear la query SQL resultante
	RS-009	Mostrar la query SQL resultante
	RS-010	Abrir conexión a la base de datos
	RS-011	Recuperar información de la base de datos con la query SQL
	RS-012	Formatear los datos recuperados
	RS-013	Mostrar los datos consultados por el usuario
	RS-014	Aplicar modelo E/R en la base de datos
	RS-015	Unicidad de nombres de tablas de la base de datos
	RS-016	Unicidad de campos de una misma tabla
	RS-017	Unicidad de identificación
Requisitos no funcionales de software	RS-018	Módulo de traducción de la consulta del usuario no complejo
	RS-019	Adhesión a recomendaciones internacionales
	RS-020	Diseño descrito documentariamente
	RS-021	Documentar cambios en la ILN

Fuente: Análisis de las entrevistas con los usuarios participantes

#### 4.1.1.5. Métrica de satisfacción de requisitos de usuario

Esta medición se hace a través de los factores de calidad de McCall, mostrados en Pressman (2010). Los resultados se

obtuvieron por medio del Anexo 6 y se resumen en el Cuadro 6, con un rango entre 0-bajo y 10-alto.

**Cuadro 6.** Puntuación de factores de calidad según usuarios del software

<b>Factores de calidad</b>	<b>Promedio (<math>m_i</math>)</b>	<b>Factores de calidad</b>	<b>Promedio (<math>m_i</math>)</b>
Exactitud	9.00	Instrumentación	8.06
Compleción	8.16	Modularidad	7.61
Concisión	7.58	Operatividad	7.94
Consistencia	7.55	Seguridad	7.84
Tolerancia al error	8.19	Auto documentación	7.71
Eficiencia de ejecución	7.65	Simplicidad	8.90
Capacidad de expansión	6.32	Trazabilidad	9.16
Generalidad	7.68	Formación	8.84

Fuente: Anexo 6. Validación de software por parte de los usuarios

Cada factor de calidad se calcula a través de la expresión:

$$F_q = (c_1 \times m_1) + (c_2 \times m_2) + \dots + (c_n \times m_n)$$

Donde,

$F_q$  : Es un factor de calidad

$c_n$  : Coeficiente de regresión

$m_n$  : Métricas que afectan al factor de calidad

De otra parte, el cálculo de los coeficientes de regresión utiliza:

$$c_n = \frac{1}{\#métricas_n}$$

Donde,  $\#métricas_n$  representa la cantidad de factores de calidad que afectan a la métrica.

El cuadro que sigue muestra el cálculo de cada métrica de McCall:

**Cuadro 7.** Cálculo de los factores de calidad

<b>Métricas McCall</b>	<b>Factores de calidad</b>	<b><math>m_i</math></b>	<b><math>c_i</math></b>	<b><math>c_i \times m_i</math></b>	<b><math>F_q</math></b>	<b>%</b>
Corrección ( <i>co</i> )	Compleción	8.16	0.33	2.72	8.29	82.90
	Consistencia	7.55	0.33	2.52		
	Trazabilidad	9.16	0.33	3.05		
Fiabilidad ( <i>fi</i> )	Exactitud	9.00	0.20	1.80	8.25	82.50
	Consistencia	7.55	0.20	1.51		
	Tolerancia a errores	8.19	0.20	1.64		
	Modularidad	7.61	0.20	1.52		
	Simplicidad	8.90	0.20	1.78		
Eficiencia ( <i>ef</i> )	Concisión	7.58	0.33	2.53	7.72	77.20
	Eficiencia de ejecución	7.65	0.33	2.55		
	Operatividad	7.94	0.33	2.65		
Integridad ( <i>in</i> )	Instrumentación	8.06	0.50	4.03	7.95	79.50
	Seguridad	7.84	0.50	3.92		
Flexibilidad ( <i>fl</i> )	Concisión	7.58	0.14	1.08	7.62	76.2
	Consistencia	7.55	0.14	1.08		
	Capacidad de expansión	6.32	0.14	0.90		
	Generalidad	7.68	0.14	1.10		
	Modularidad	7.61	0.14	1.09		
	Auto documentación	7.71	0.14	1.10		
	Simplicidad	8.90	0.14	1.27		

Capacidad de pruebas (cp)	Instrumentación	8.06	0.25	2.02	8.07	80.70
	Modularidad	7.61	0.25	1.90		
	Auto documentación	7.71	0.25	1.93		
	Simplicidad	8.90	0.25	2.23		
Reusabilidad (re)	Generalidad	7.68	0.33	2.56	7.67	76.70
	Modularidad	7.61	0.33	2.54		
	Auto documentación	7.71	0.33	2.57		
Usabilidad (us)	Operatividad	7.94	0.50	3.97	8.39	83.90
	Facilidad de formación	8.84	0.50	4.42		

Fuente: Basado en el Cuadro 6

Para la actividad de mantenimiento se utiliza el índice de madurez de software (*IMS*) que se calcula con:

$$IMS = \frac{[M_r - (F_a + F_c + F_d)]}{M_r}$$

Donde,

$M_r$  : Número de módulos en la versión actual

$F_c$  : Número de módulos en la versión actual que se han cambiado

$F_a$  : Número de módulos en la versión actual que se han añadido

$F_d$  : Número de módulos en la versión anterior que se han borrado en la versión actual

En el caso de la ILN se tiene que:  $M_r = 22, F_a = 0, F_c = 4, F_d = 0 \therefore$

$IMS = 0.8182$ . Valor cercano a 1.0, lo que indica la proximidad del producto a su estabilización.

Para la puntuación final de calidad de software ( $Pcs$ ), o sea, el grado de satisfacción de la aplicación, se promediaron los resultados de las métricas de McCall.

$$Pcs = \frac{co + fi + ef + in + fl + cp + re + us}{8}$$

$$Pcs = \frac{8.29 + 8.25 + 7.72 + 7.95 + 7.62 + 8.07 + 7.67 + 8.39}{8} \cong 7.995$$

El cálculo anterior, en una escala de 0 a 10, representa un grado de satisfacción de requerimientos enteramente aceptable.

**4.1.1.6. Identificación de actores**

**Cuadro 8.** Detalle de los actores que interactúan con la ILN

Tipo de actor	Perfil	Funciones
Usuario (directo/final) u Operario	Alumno (o estudiante)	<ul style="list-style-type: none"> <li>- Utiliza directamente la ILN</li> <li>- Habilita una conexión a la base de datos mediante su identificación</li> <li>- Escribe las consultas para obtener información de la base de datos</li> <li>- Visualiza los resultados obtenidos</li> </ul>

Fuente: Entrevistas con los usuarios participantes

**4.1.1.7. Análisis de los casos de uso**

Esta parte describe la forma en que el usuario interactúa con el software, es decir, la funcionalidad visible para el usuario (el comportamiento del software de acuerdo a los requisitos funcionales, véase el Cuadro 42 al final del Anexo 5):

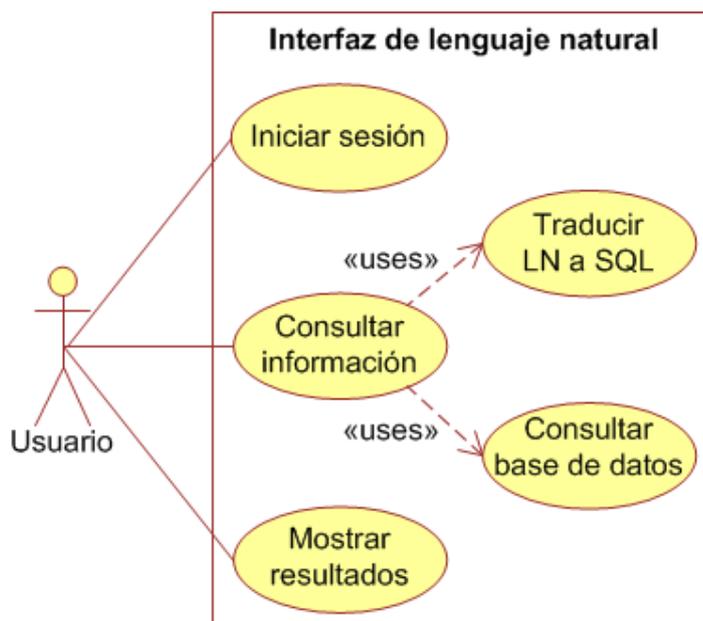


Figura 3. Diagrama de casos de uso

4.1.1.8. Realización de los casos de uso

Cuadro 9. Descripción del caso de uso “Iniciar sesión”

Caso de uso	Iniciar sesión
Descripción:	Habilita la ILN para consultar la base de datos
Conjeturas:	- El Usuario conoce previamente el nombre y contraseña para iniciar la ILN (que son los mismos de la base de datos y que no necesita registrar antes)
Condiciones previas:	- El Usuario inició el funcionamiento de la ILN
Condiciones resultantes:	- El Usuario visualiza la pantalla principal de la ILN con todos sus elementos activados
Pasos:	1. El Usuario escribe su nombre en una caja de texto vacía del formulario de ingreso 2. El Usuario escribe su contraseña en una caja de texto vacía del mismo formulario de ingreso 3. Hace clic en el botón correspondiente o presiona Enter para verificar los datos ingresados (E-1)



	<p>4. El sistema valida el nombre y la contraseña (E-2)</p> <p>5. El sistema oculta el formulario de ingreso</p>
Condición excepcional o alternativa:	<p>E-1 Si alguna de las cajas de texto está vacía se muestra un mensaje de advertencia y se reinicia el caso de uso</p> <p>E-2 Si alguno de los datos introducidos es incorrecto se muestra un mensaje de error, y el caso de uso se reinicia</p>
Actor que inicia:	Usuario
Actor beneficiado:	Usuario

**Cuadro 10.** Descripción del caso de uso “Consultar información”

<b>Caso de uso</b>	<b>Consultar información</b>
Descripción:	El Usuario consulta la base de datos para recuperar información de ella, enunciando dicha consulta por escrito mediante lenguaje natural, términos clave o una mezcla de ambos
Conjeturas:	- El usuario quiere usar la ILN para saber cómo se escribiría una <i>query</i> SQL
Condiciones previas:	- El usuario inició sesión correctamente
Condiciones resultantes:	- Genera una query SQL de recuperación de información
Pasos:	<p>1. El Usuario escribe la consulta en el elemento de la interfaz creado para tal efecto</p> <p>2. Hace clic en el botón correspondiente o presiona Enter para empezar el procesamiento de la consulta (E-1)</p> <p>3. Incluye caso de uso “Traducir LN a SQL” (E-2)</p> <p>4. Incluye el caso “Consultar base de datos”</p>



Condición excepcional o alternativa:	E-1 Si el Usuario no escribe nada en la caja de texto no se procesa ninguna consulta y se reinicia el caso de uso E-2 Si no se puede transformar la consulta sólo se muestra un mensaje de error y se termina el caso de uso
Actor que inicia:	Usuario
Actor beneficiado:	(Ninguno)

**Cuadro 11.** Descripción del caso de uso “Traducir LN a SQL”

<b>Caso de uso</b>	<b>Traducir LN a SQL</b>
Descripción	Transforma la consulta escrita por el Usuario a una query SQL semánticamente equivalente
Conjeturas:	- El Usuario escribe una consulta en lenguaje humano, términos clave o ambos; inteligible y gramaticalmente correcta - El Usuario conoce ciertas reglas de antemano para pre formatear su consulta
Condiciones previas:	- El Usuario escribe, y pre formatea debidamente su consulta según necesite
Condiciones resultantes:	- Se genera una cadena de caracteres que representa la query SQL de la consulta hecha
Pasos:	1. La ILN procesa la consulta hecha por el Usuario enviándola a un módulo traductor 2. El módulo traductor pasa la consulta al analizador morfológico que devuelve la consulta normalizada 3. El módulo remite la consulta normalizada al analizador sintáctico, el cual retorna la estructura sintáctica (E-1) 4. El módulo traductor da la estructura sintáctica



	<p>al analizador semántico. Este regresa una “pseudo query SQL”</p> <p>5. El módulo transforma, mediante el analizador pragmático, la pseudo query SQL en una sentencia SQL válida y la devuelve para ser procesada</p>
Condición excepcional o alternativa:	E-1 Si la estructura sintáctica es incorrecta se termina el caso de uso
Actor que inicia:	(Ninguno)
Actor beneficiado:	(Ninguno)

**Cuadro 12.** Descripción del caso de uso “Consultar base de datos”

<b>Caso de uso</b>	<b>Consultar base de datos</b>
Descripción:	Recibe una query SQL y consulta con ella la base de datos
Conjeturas:	<ul style="list-style-type: none"> <li>- La ILN envía <i>queries</i> SQL a la base de datos</li> <li>- La base de datos está cargada correctamente con la información sobre la cual se hace la consulta</li> </ul>
Condiciones previas:	<ul style="list-style-type: none"> <li>- La ILN tiene una conexión activa hacia la base de datos</li> <li>- La query SQL que se recibe/envía es correcta</li> </ul>
Condiciones resultantes:	- Devuelve la información recuperada de la base de datos por medio de la query SQL en forma de <i>array</i>
Pasos:	<ol style="list-style-type: none"> <li>1. La ILN procesa la query SQL transmitiéndola al gestor de la base de datos</li> <li>2. El gestor envía la query SQL al manejador de la base de datos</li> <li>3. El manejador consulta la base de datos en</li> </ol>

	<p>base a la query</p> <p>4. La base de datos retorna el resultado de la consulta al manejador (E-1)</p> <p>5. El manejador devuelve el resultado al gestor en forma de array (E-2)</p> <p>6. El gestor reenvía el array para procesarse</p>
Condición excepcional o alternativa:	<p>E-1 Si la query SQL es errónea genera un mensaje y se termina el proceso</p> <p>E-2 Si ningún dato satisface la consulta se devuelve un array vacío</p>
Actor que inicia:	(Ninguno)
Actor beneficiado:	(Ninguno)

**Cuadro 13.** Descripción del caso de uso “Mostrar resultados”

<b>Caso de uso</b>	<b>Mostrar resultados</b>
Descripción:	Despliega los resultados de consultar la base de datos
Conjeturas:	<ul style="list-style-type: none"> <li>- La ILN emite mensajes de error y advertencia</li> <li>- La ILN aplica el formato y estilo predefinidos para queries SQL y tablas HTML</li> </ul>
Condiciones previas:	- El procesador de la consulta recibe un array proporcionado por el gestor de base de datos
Condiciones resultantes:	<ul style="list-style-type: none"> <li>- Se visualiza la query SQL generada por la ILN al traducir la consulta del usuario en SQL debidamente formateada</li> <li>- Se visualiza en forma de tabla HTML la información recuperada de la base de datos con la query generada</li> </ul>
Pasos:	1. Se envía el array recibido del gestor de base de datos, junto con la query SQL, a la interfaz de usuario



	<p>2. La interfaz de usuario formatea la query para su presentación posterior (E-1)</p> <p>3. La interfaz de usuario convierte el array a una tabla HTML (E-2)</p> <p>4. Se muestra la query en un espacio de la pantalla principal, específicamente preparado para ello</p> <p>5. Se muestra la tabla HTML en un espacio definido para ese propósito</p>
Condición excepcional o alternativa:	<p>E-1 Si la query es una cadena vacía se muestra un mensaje de advertencia y se termina el caso de uso</p> <p>E-2 Si el array está vacío se muestra un mensaje de advertencia y se termina el caso de uso</p>
Actor que inicia:	(Ninguno)
Actor beneficiado:	Usuario

**4.1.1.9. Identificación de las clases involucradas**

En base al vocabulario del dominio del problema se procede a identificar y describir las principales clases participantes por estereotipos (Control, Entidad y Límite). En el cuadro siguiente se exhiben las que manejan la comunicación entre el entorno y lo interno del sistema, o sea, las que suministran las interfaces con otros usuarios o (sub)sistemas:

**Cuadro 14.** Identificación de las clases de límite de la aplicación

Clase	Atributos	Responsabilidades
Gestor de base de	<ul style="list-style-type: none"> <li>- SQL</li> <li>- Datos</li> </ul>	1. Envía el SQL al manejador de la base de datos

datos	recuperados	2. Devuelve datos recuperados por el Manejador de la base de datos
IU	- Nombre - Contraseña - Resultados	1. Presenta los formularios 2. Lee las entradas del Usuario 3. Se comunica con el Gestor de usuario para habilitarlo 4. Visualiza advertencias, errores o datos para el Usuario
NI2Sql	- Consulta - SQL	1. Convierte consulta a SQL 2. Devuelve el SQL
Procesa consulta	- Consulta - SQL - Datos recuperados	1. Recibe consulta 2. Envía consulta para obtener SQL 3. Envía SQL al Gestor de base de datos para obtener información 4. Devuelve SQL y datos recuperados

Las clases de control presentan no sólo la funcionalidad, sino también la dinámica de los casos de uso, de forma que coordinan los eventos a realizar para una conducta específica:

**Cuadro 15.** Identificación de las clases de control de la aplicación

Clase	Atributos	Responsabilidades
Autómata	- Consulta normalizada - Tabla de transición	1. Recorre cada elemento de la consulta normalizada de acuerdo a una tabla de transición
Combinar tablas	- Ruta de grafo	1. Crea combinaciones entre tablas de acuerdo al Grafo
Consulta SQL	- Listas de campos, tablas,	1. Agrega campo, tabla, condición u orden de

	condiciones, ordenamientos	presentación de los datos
Gestor de usuario	- Identificación de usuario	1. Activa Usuario
Grafo	- Representación de la base de datos con un grafo	1. Halla la ruta más óptima entre las relaciones de las tablas
Manejador MySQL	- Servidor - Usuario - Contraseña	1. Abre y cierra una conexión a la base de datos 2. Consulta una tabla de la base de datos
Morfológico	- Consulta - Consulta normalizada	1. Parsea y tokeniza consulta 2. Reemplaza acentos y puntuaciones 3. Condensa literales compuestos 4. Normaliza y obtiene el lema de cada palabra
Pragmático	- Pseudo SQL - Columnas - Tablas - Condiciones - Combinaciones	1. Interpreta el significado del Pseudo SQL 2. Invoca combinaciones 3. Crea un nuevo SQL
Semántico	- Estructura sintáctica - Consulta	1. Interpreta elementos de la consulta para darle un significado, y crea un Pseudo SQL que necesita ser refinado (en muchos casos)
Significado	- Consulta	1. Obtiene consulta 2. Filtra consulta 3. Interpreta comodines

Sintáctico	- Consulta normalizada	1. Activa la clase Autómata
------------	------------------------	-----------------------------

Las clases entidad almacenan en forma persistente los datos de la aplicación:

**Cuadro 16.** Identificación de las clases de entidad de la aplicación

Clase	Atributos	Responsabilidades
Base de datos	- Manejador	1. Consulta una tabla de la base de datos, creando el manejador más apropiado (MySQL)
Gestor usuario	- Nombre - Contraseña	1. Guarda y/o compara el nombre y contraseña introducidos por el Usuario y los utiliza para conectar la base de datos

#### 4.1.1.10. Métrica de puntos de función

Esta técnica mide el tamaño funcional del software, desde el punto de vista del cliente, fuera de la tecnología empleada para su construcción (Pressman, 2010), y se usa para estimar el esfuerzo, duración y costo del proyecto software. Se computa determinando cinco características de dominio de información con complejidades asignadas a cada una individualmente: Entradas de usuario, Salidas para éste, Consultas externas, Archivos (lógicos internos), e Interfaces externas.

Como se ve, se contaron los requisitos pertenecientes a cada una de las características anteriores. Cada conteo, se multiplica por un

valor estándar que encarna la complejidad subjetiva. La suma de esas multiplicaciones son los Puntos de Función sin Ajustar (*PFSA*):

**Cuadro 17.** Factores de ponderación

Tipo\Complejidad	Cuenta	Simple	Medio	Complejo	Total
Entradas/usuario	<u>2</u>	<u>3</u> PF	4 PF	6 PF	6
Salidas/usuario	<u>2</u>	<u>4</u> PF	5 PF	7 PF	8
Peticiones/usuario	<u>1</u>	3 PF	<u>4</u> PF	6 PF	4
Archivos	<u>6</u>	7 PF	10 PF	<u>15</u> PF	90
Interfaces externas	0	5 PF	7 PF	10 PF	0
Puntos de función sin ajuste ( <i>PFSA</i> )					108

Para obtener Puntos de Función Ajustados (*PFA*) se utilizan valores de ajuste de la complejidad ( $F_i$ ), a los cuales se les asigna empíricamente un puntaje normado (0-es no importante o aplicable, 1-incidental, 2-moderado, 3-medio, 4-significativo, 5-absolutamente esencial).

**Cuadro 18.** Valores de ajuste de la complejidad

$F_i$	Puntaje
¿Hay herramientas que ayudan a transferir información?	0
¿Cómo se manejan los datos distribuidos?	0
¿El tiempo de respuesta es requerido por el usuario?	1
¿Qué tanto se usa el hardware donde se ejecuta?	1
¿Cuán frecuentemente se ejecutan transacciones?	3
¿Qué porcentaje de transacciones en línea hay?	0
¿Se diseñó pensando en la eficiencia del usuario final?	3
¿Cuántos archivos se actualizan en línea?	0
¿Tiene mucho procesamiento lógico o matemático?	3
¿Cumple una o muchas necesidades del usuario?	4
¿Qué tan difícil es la conversión de datos e instalación?	5



¿Es efectivo el inicio, respaldo y recuperación?	0
¿Se creó para ser instalado en múltiples sitios?	2
¿Se creó para facilitar su mantenimiento?	3
Factor de ajuste ( $\sum F_i$ )	25

Fuente: Basado en Pressman (2010)

La suma de estos puntajes da el factor de ajuste ( $\sum F_i$ ). Ahora bien, aplicando la siguiente fórmula:

$$PFA = PFSA \times (0,65 + (0,01 \times \sum F_i))$$

Y reemplazando los valores se tiene que:

$$PFA = 108 \times (0,65 + (0,01 \times 25)) = 97.2 \cong 97$$

El tamaño de la aplicación es de 97 puntos de función. Este resultado sirve para estimar el esfuerzo requerido para desarrollar el proyecto de software en meses/hombre por punto de función. Ahora, considerando la siguiente tabla:

**Cuadro 19.** Evaluación de horas y líneas de código por lenguaje

Lenguaje	Horas PF promedio	Líneas de código por PF
Ensamblador	25	300
COBOL	15	100
4ta Generación	8	20

Fuente: IFPUG

Utilizando la última fila en la siguiente fórmula:

$$H/H = PFA \times \text{Horas PF Promedio}$$

$$H/H = 97 \times 8 = 776$$

Las horas hombre ( $H/H$ ) para hacer el software es de 776. Si se desarrolla el software de lunes a sábado cuatro horas diarias con un trabajador se tendrán 194 días de trabajo equivalente 5.39 meses. En cuanto al costo, aplicando la siguiente fórmula:

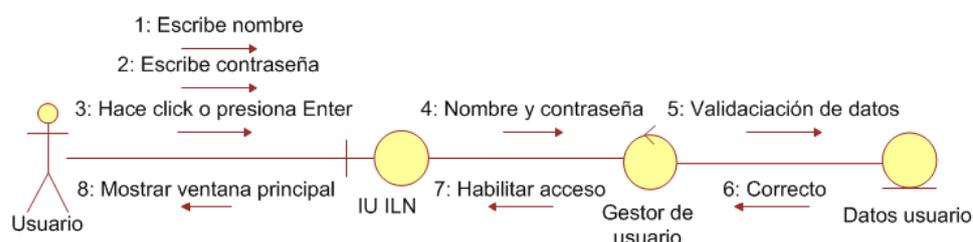
$$\text{Costo} = (\text{desarrolladores} \times \text{duración} \times \text{sueldos}) + \text{otros}$$

Donde, *desarrolladores* es uno (01), *duración* (del proyecto) es 5.39, *sueldos* igual a cero (0), y *otros* (gastos) se calculan en S/.500.00, el resultado estimado del costo del proyecto es S/.500.00. Estas estimaciones de costo y tiempo resultan enteramente aceptables.

#### 4.1.2. Diseño

##### 4.1.2.1. Estructura de clases

Los diagramas de colaboración expresan las interacciones de los objetos involucrados en los casos de uso, como se aprecia en los siguientes diagramas.



**Figura 4.** Diagrama de colaboración "Iniciar sesión"

La ilustración previa exhibe la realización del caso de uso que permite el acceso a la ILN. La siguiente figura enseña la colaboración de las clases involucradas en procesar la consulta:

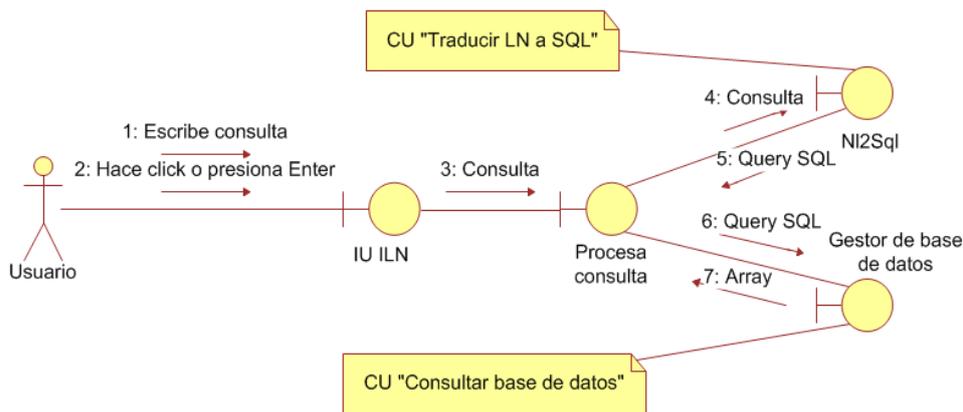


Figura 5. Diagrama de colaboración "Consultar base de datos"

La imagen que viene describe las clases intervinientes para realizar la traducción del LN a SQL:

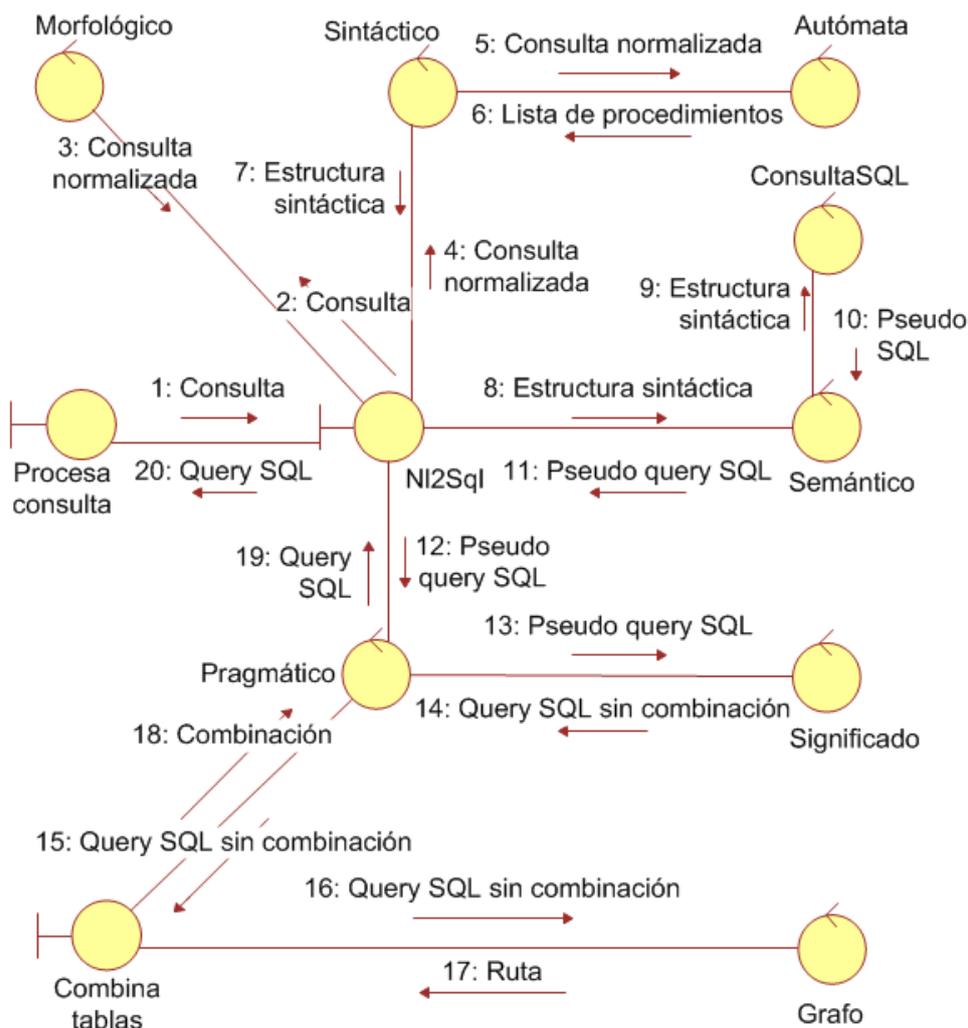
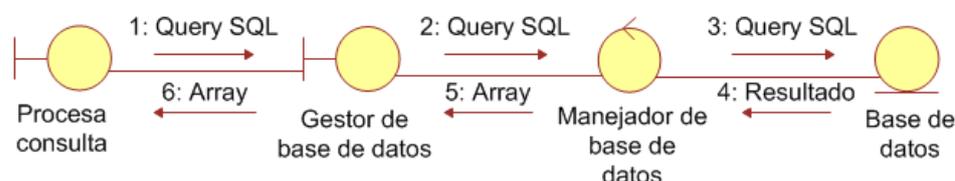


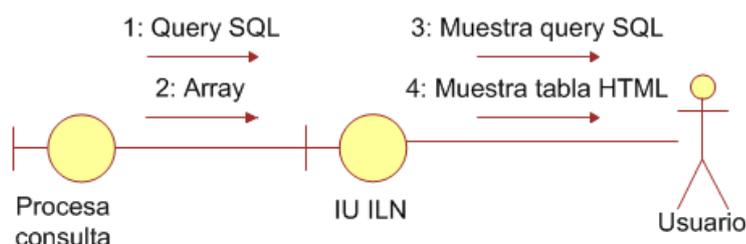
Figura 6. Diagrama de colaboración "Traducir LN a SQL"

También se ve la comunicación de Procesar consulta con el gestor de base de datos:



**Figura 7.** Diagrama de colaboración “Consultar base de datos”

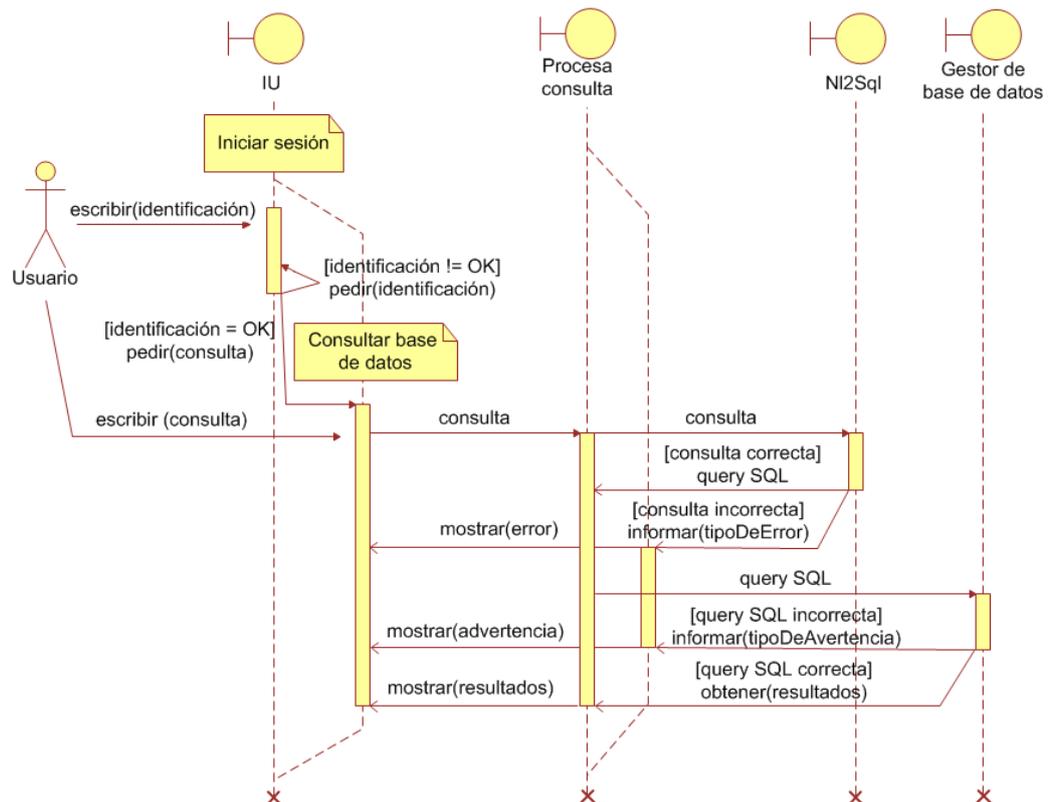
Por último aparece la colaboración que muestra los resultados al usuario, después de realizar la traducción y la consulta a la base de datos:



**Figura 8.** Diagrama de colaboración “Visualizar resultados”

La interacción de los objetos involucrados en la solución (p.ej.: dónde empieza o termina, o en qué momento se produce, una determinada acción), se manifiesta mediante un “diagrama de secuencia”, proporcionado por UML, y que suele representar los diagramas de casos de uso, como en la siguiente imagen, que se describe así:

1. La IU está en inicio de sesión. El Usuario se identifica (ingresa su nombre y contraseña) por medio de la IU.
2. La IU verifica si la identificación del Usuario es correcta.



**Figura 9.** Diagrama de secuencias principal de la ILN

1. Si la identificación es incorrecta, entonces se repite todo el proceso anterior.
2. De lo contrario, la IU entra en estado de consulta, abre la ventana principal y muestra un Formulario de Consulta.
3. El Usuario escribe en el Formulario de Consulta la información que quiere recuperar de la base de datos por medio de la ILN.
4. La consulta del usuario se pasa al Procesador de la Consulta.
5. El Procesador envía la consulta al NI2Sql.
6. El NI2Sql verifica si puede traducirse la consulta a SQL.

7. Si no es traducible, entonces se pasa al Procesador un mensaje de error. Este pasa el mensaje a la IU que muestra el mensaje al Usuario.
8. Si la consulta es traducible, entonces se transforma la consulta en una query SQL y se la devuelve al Procesador.
9. El Procesador transmite la query al Gestor de base de datos que la emplea para consultar la base de datos.
10. Si la query es incorrecta, o sea, genera errores en la base de datos o produce un resultado vacío, entonces se retorna un mensaje de advertencia al Procesador, y esta a su vez lo reenvía a la IU para ser mostrado al Usuario.
11. Si la query es correcta, el Gestor retorna un resultado que se pasa al Procesador. Este da el resultado a la IU, junto con la query para que sean presentados.

El estado de consulta puede continuar indefinidamente hasta que se abandona la ILN.

#### **4.1.2.2. Estructura arquitectónica**

Se utiliza la llamada Programación por Capas, que permite desarrollar aplicaciones en varios estratos escalables, cada cual con una tarea simple. Así, si sobrevienen cambios se les adecua en la capa correspondiente sin modificar otros niveles. En esta investigación se utiliza el diseño más común, que es el de tres capas (Presentación, Negocio y Datos), el cual se ve así:



**Figura 10.** Arquitectura de tres capas

#### 4.1.2.3. Diseño de módulos

Una capa puede coincidir con el concepto de paquete, y éste con el de carpeta en el disco físico, de forma que la ILN se ve como en la Figura 11. En lo sucesivo se hará mención de muchos paquetes que tengan un nombre similar al de sus clases, mas no deben ser confundidos; los paquetes principales tienen nombres resaltados, a diferencia de otro tipo de elementos.

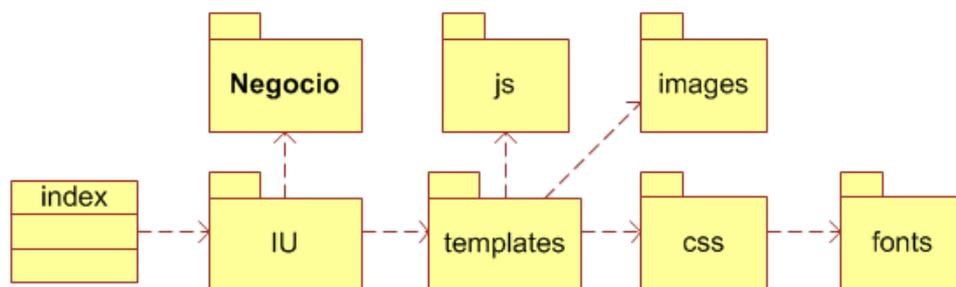


**Figura 11.** Organización interna de las capas de la ILN

La anterior ilustración exhibe la clase *index* aislada, por estrategia, de cualquier paquete y que es donde empieza a ejecutarse la ILN.

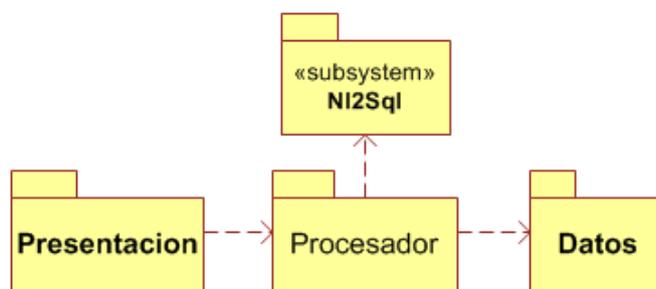
Las flechas punteadas representan llamadas a diferentes elementos. *index* llama a la clase IU del sub paquete homónimo guardado en *Presentacion*. Por su lado, algún elemento de IU, solicita la asistencia de otros paquetes, y así sucesivamente. Esta

ramificación entre llamadas permite un mejor orden y mantenimiento del software.



**Figura 12.** Estructura interna del paquete Presentacion

La capa o paquete Presentacion muestra la interfaz gráfica, para recibir las solicitudes del, y presentar los resultados al, usuario. Esta capa se comunica únicamente con la de Negocio.

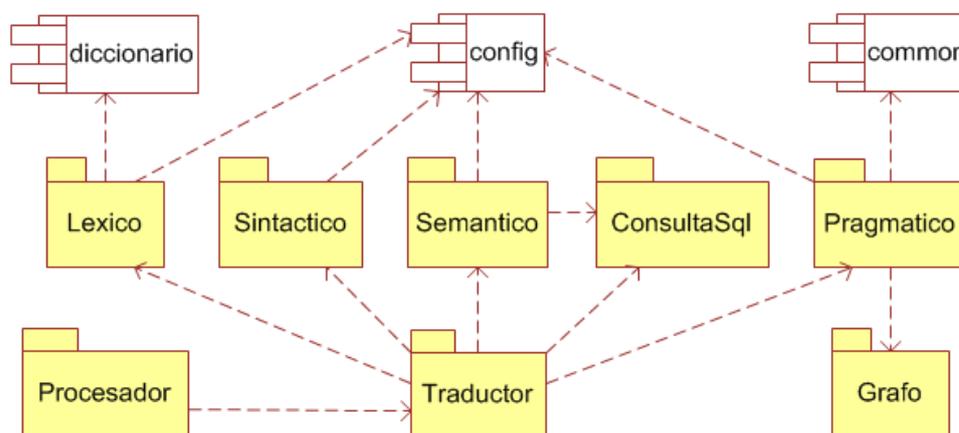


**Figura 13.** Estructura interna del paquete Negocio

En Negocio reside lo que hará la ILN. Así, algún elemento de Presentacion apela a otro oculto en Procesador para transmitirle una petición del usuario. Procesador la resuelve de acuerdo a la lógica del programa y pide la asistencia de NI2Sql. Finalmente, se comunica con Datos.

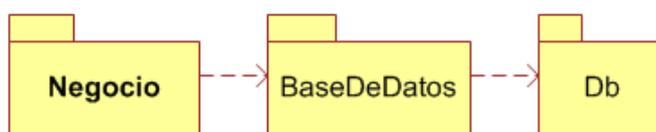
Nótese que se mencionó la palabra “oculto”, como referencia del concepto de “ocultación de información”.

El subsistema NI2Sql (*Natural Language To SQL*) es el núcleo de la ILN, razón por la que se describe su composición interna:



**Figura 14.** Estructura interna del sub sistema NI2Sql

Cada sub paquete (Morfologico, Sintactico, etc.) incluye las clases ya vistas en 4.1.1.9. Además, se aprecian tres componentes libres que realizan tareas más simples: *common*, con métodos de propósito general; *config*, que define la configuración; y *diccionario*, que guarda el conjunto de palabras usadas para la traducción.

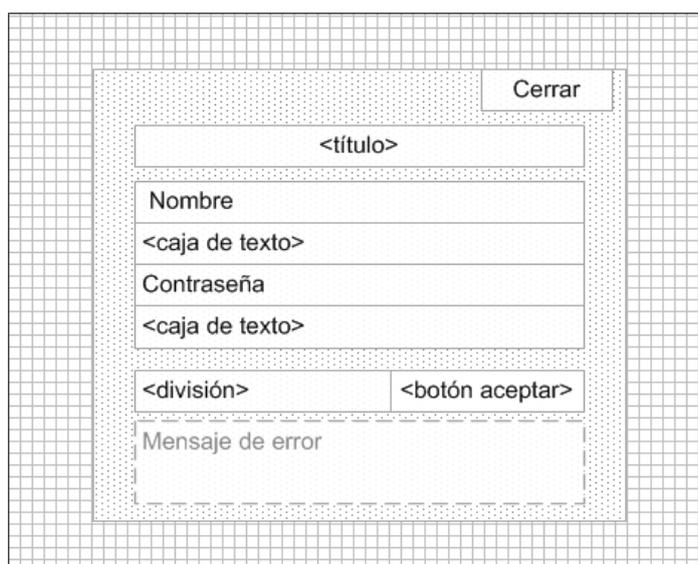


**Figura 15.** Estructura interna del paquete Datos

Por su lado, Datos, realiza la conexión con la base de datos. Está formado por un gestor de base de datos que recibe solicitudes de la capa de Negocio para recuperar información. La información recuperada se devuelve a Negocio.

#### 4.1.2.4. Diseño de la estructura o *layout* de la interfaz gráfica

La interfaz gráfica de la ILN intenta ser simple, atractiva y útil. Está construida en PHP y aplica el modelo de cajas HTML. Al inicio, se pide al usuario identificarse, para ello hay un formulario específico:

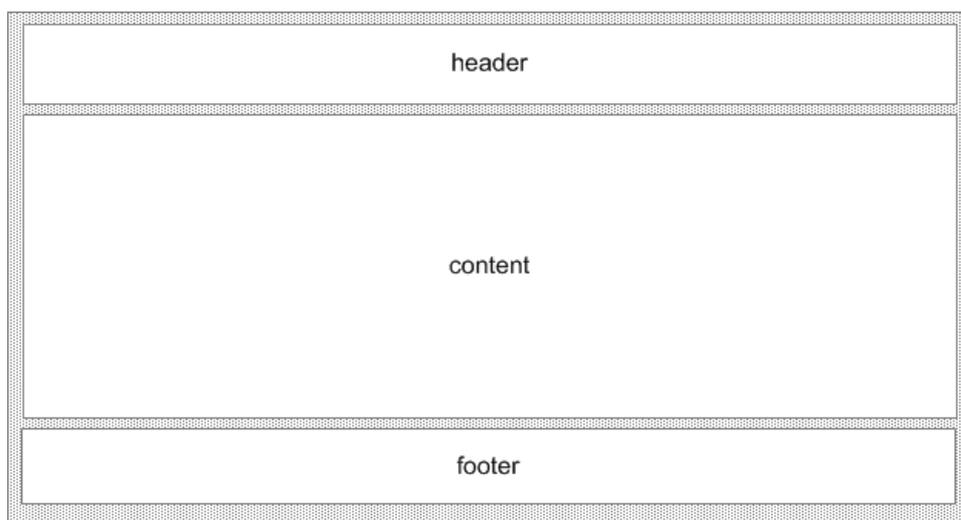


**Figura 16.** Esquema del diseño del formulario de identificación

La figura anterior enseña una ventana modal sobre una pantalla ensombrecida. Dicha ventana tiene un título de bienvenida, y espacios debidamente indicados para ingresar el nombre y contraseña de la base de datos.

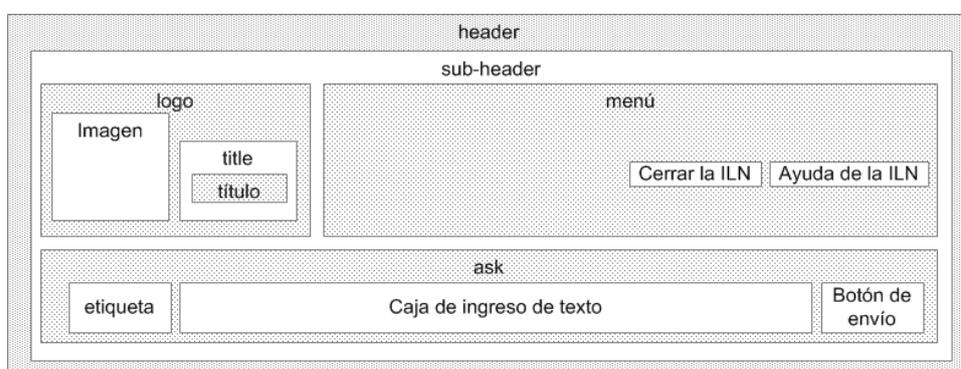
Tanto "Cerrar" como "botón aceptar" son botones que, respectivamente, cancelan o verifican el ingreso de datos del usuario. En caso de una incorrección en los datos entrados, se muestra un mensaje de error (elemento con líneas punteadas).

Si la identificación es correcta el usuario ve la ventana principal, cuya estructura general se ve como sigue:



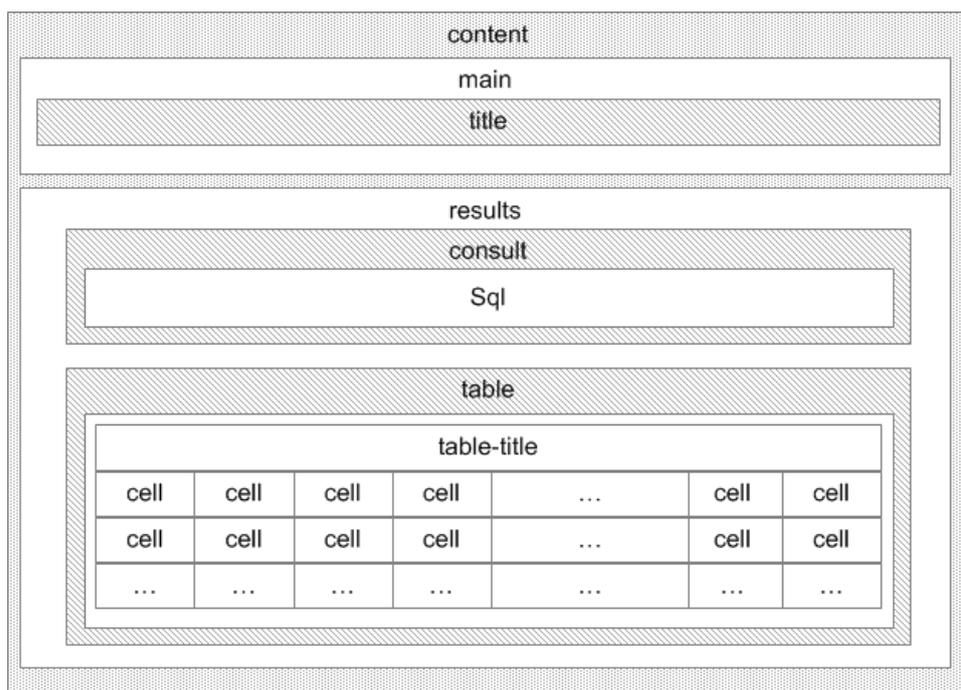
**Figura 17.** Diseño de la pantalla principal de la ILN

En *header* está la descripción de la ILN, un menú y un espacio para consultar. El menú tiene los botones Cerrar (que termina la sesión de trabajo) y Ayuda (que abre un archivo de asistencia). El espacio de consulta (*ask*) es el Formulario de consulta como se ve en seguida:



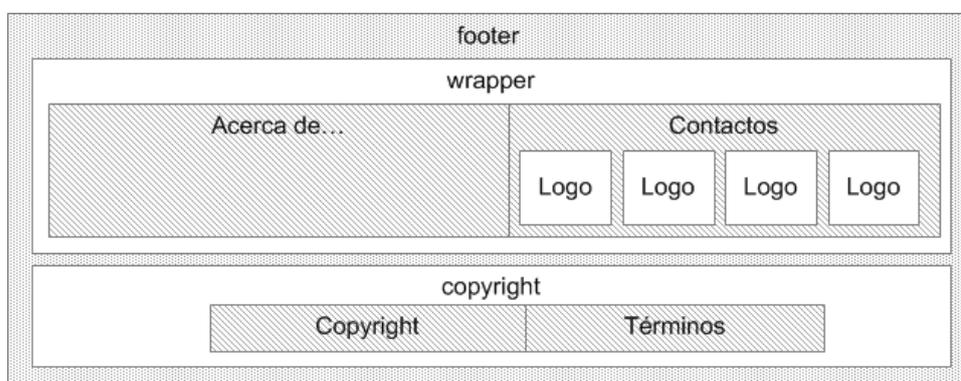
**Figura 18.** Esquema de la cabecera de la pantalla principal

Por otro lado, *content* publica los resultados. Los espacios *consult* y *table* enseñan, equitativamente, la query producida (al traducir la consulta del usuario al lenguaje SQL) y la información recuperada (de la base de datos) con dicha query:



**Figura 19.** Modelo de cajas de la zona central de la pantalla principal de la ILN

El *footer* comprende los términos de uso, datos del desarrollador y la información de contacto con éste.

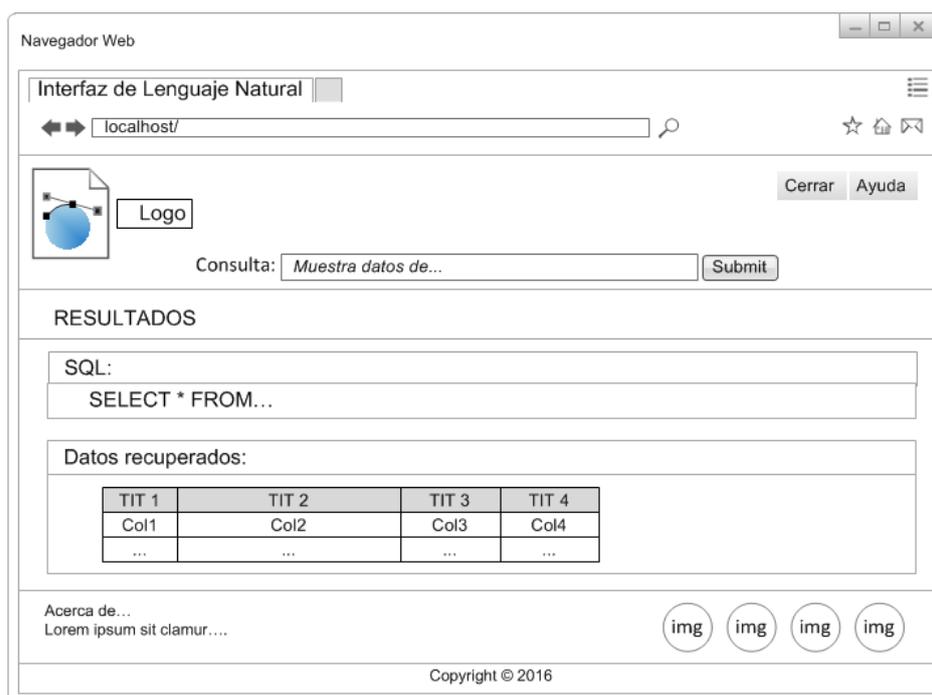


**Figura 20.** Esquema de la pantalla principal de la ILN

Para tener una idea más cabal del diseño completo de la interfaz, se ven las salidas aproximadas de las imágenes anteriores en un navegador web con soporte HTML5 y CSS3.



**Figura 21.** Diagrama reticular de la ventana de identificación



**Figura 22.** Diagrama reticular de la pantalla principal de la ILN

#### 4.1.2.5. Diseño de datos

La base de datos que usa la ILN es la del Registro de Alumnos del Instituto (Rosado, 2016). Los números de cada tabla pueden variar dado que fueron aumentados para orientar el diseño del dígrafo.

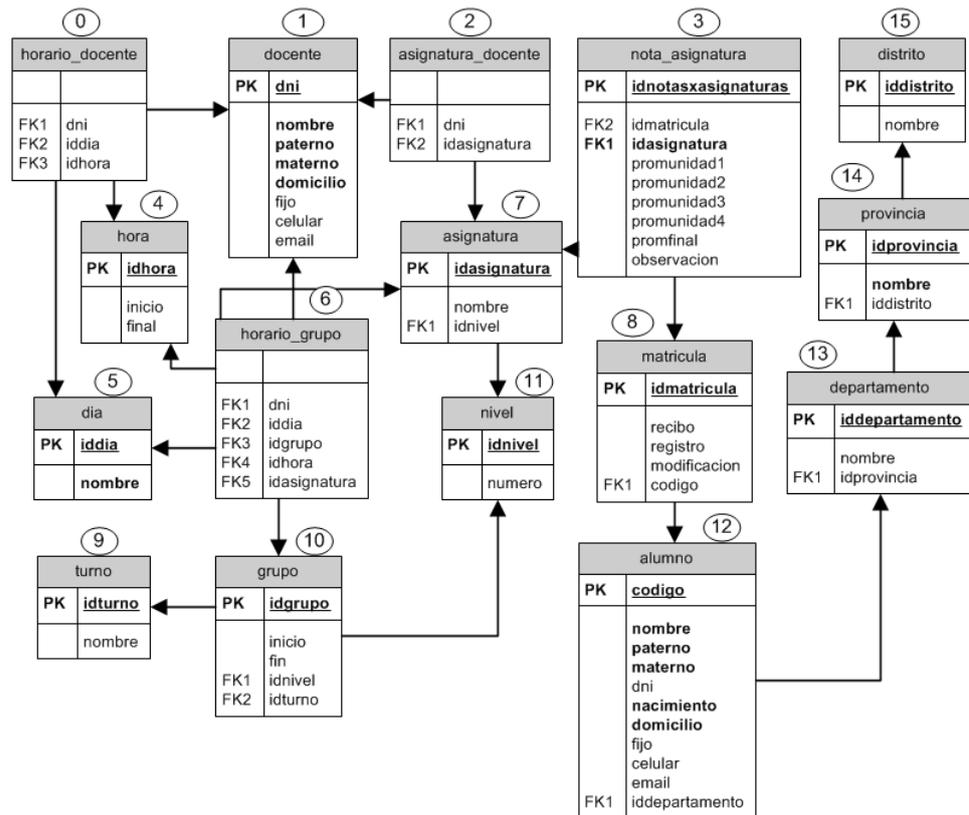


Figura 23. Diseño de la base de datos de la ILN

4.1.2.6. Diseño del grafo

El dígrafo se basa en la figura anterior, su importancia radica en que recorriéndolo se crean combinaciones entre tablas, es decir, se determina el orden y la forma en que éstas se vinculan.

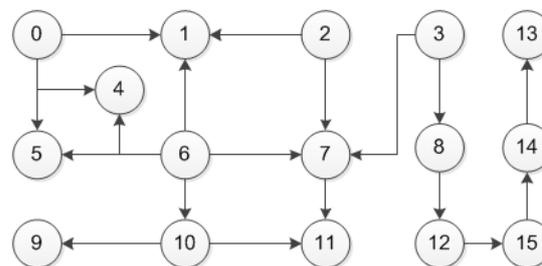


Figura 24. Grafo direccional de búsqueda



**4.1.2.8. Métrica de diseño**

Dentro del paquete de Presentación no existen muchas dependencias, lo que se manifiesta en su complejidad estructural. Igualmente, la complejidad de datos hace notar que las variables de entrada/salida son sencillas:

**Cuadro 20.** Cálculo de complejidad en el paquete Presentación

Módulo i (Presentación)	Expansión del módulo $f_{out}(i)$	Complejidad estructural $S(i) = f_{out}^2(i)$	$v(i)$	Complejidad de datos $D(i) = \frac{v(i)}{f_{out}(i)+1}$	Complejidad del sistema $C(i) = S(i) + D(i)$
inicio.html	2	4	3	1	5
general.html	1	1	4	2	3
style.css	0	0	0	0	0
complems.js	0	0	5	5	5
recuperar.php	2	4	5	1.7	5.7
IU.php	3	9	6	1.5	10.5
Subtotal		18		11.2	29.2
Complejidad relativa					9.7

El cuadro que viene evidencia que el paquete Negocio tiene una complejidad relativa de 9.16, representando una mínima cantidad de dependencias y de flujo de datos.

**Cuadro 21.** Cálculo de complejidad en el paquete Negocio

Módulo i (Negocio)	Expansión del módulo $f_{out}(i)$	Complejidad estructural $S(i) = f_{out}^2(i)$	$v(i)$	Complejidad de datos $D(i) = \frac{v(i)}{f_{out}(i)+1}$	Complejidad del sistema $C(i) = S(i) + D(i)$
Procesador.php	1	1	1	0.5	1.5
Traductor.php	2	4	2	0.7	4.7

Morfológico.php	2	4	2	0.7	4.7
Sintactico.php	1	1	2	1	2
Autómata.php	1	1	2	1	2
Semantico.php	4	16	2	0.4	16.4
Pragmatico.php	3	9	3	0.8	9.8
GeneraSql.php	2	4	12	4	8
CaracterDeSustitución.php	4	16		0	16
Grafo.php	0	0	7	7	7
common.php	0	0	3	3	3
config.php	0	0	42	42	42
diccionario.php	0	0	2	2	2
Subtotal		56		63.1	119.1
Complejidad relativa					9.16

Datos tiene una complejidad relativa de 2.5, lo que simboliza una exigua cantidad de dependencias y de entradas/salidas de datos.

Esto indica que dicho paquete es fácil de implementar, como se ve:

**Cuadro 22.** Cálculo de complejidad en el módulo Datos

Módulo i (Datos)	Expansión del módulo $f_{out}(i)$	Complejidad estructural $S(i) = f_{out}^2(i)$	$v(i)$	Complejidad de datos $D(i) = \frac{v(i)}{f_{out}(i)+1}$	Complejidad del sistema $C(i) = S(i) + D(i)$
BaseDeDatos.php	1	1	2	1	2
ManejadorMySql.php	1	1	4	2	3
Subtotal		2		3	5
Complejidad relativa					2.5

En resumen: la suma de complejidades estructural y de datos son respectivamente de 76 y 77.3, señalando la pequeña dependencia y complejidad que tienen. Aunado a eso, la complejidad del sistema

es de 153.3, con una CRS de 21.36 que es menor a 26.5; lo que indica que el modelo arquitectónico es aceptable, como sigue:

**Cuadro 23.** Complejidad del software ILN

Módulo i	Complejidad estructural $S(i) = f_{out}^2(i)$	Complejidad de datos $D(i) = \frac{v(i)}{f_{out}(i)+1}$	Complejidad del sistema $C(i) = S(i) + D(i)$	Complejidad relativa (CRS)
Presentación	18	11.2	29.2	9.7
Negocio	56	63.1	119.1	9.16
Datos	2	3	5	2.5
Total	76	77.3	153.3	21.36

#### 4.1.3. Implementación

##### 4.1.3.1. Metodología

Se emplea el paradigma de la Programación Orientado a Objetos.

##### 4.1.3.2. Herramientas

El front-end se construyó aprovechando HTML5, CSS3, JavaScript y PHP. Este último utiliza el servidor Apache, por lo que la elección obvia para el uso de todos estos elementos recae en el XAMPP 1.8.3, que también incorpora MySQL. Para el diseño de la base de datos se emplea MySQL Workbench 6.3. Como editor de código se usa Sublime Text 3, y se visualiza la ILN en el navegador Chrome 64.0 (aunque también se puede en Firefox 42.0, Opera 33.0, Safari 5.1.7, IExplorer 8, etc.)

#### 4.1.3.3. Principales algoritmos

Se presenta en forma de pseudocódigo el subsistema NI2Sql (para ver algunos detalles más consúltese el Anexo 8):

---

##### **Algoritmo 1:** Convertir LN a SQL

---

**Entrada**  $c_0$ : consulta del usuario en LN

1. *consulta normalizada*  $\leftarrow$  *analizar morfológicamente* ( $c_0$ )
  2. *consulta sintáctica*  $\leftarrow$   
*analizar sintácticamente* (*consulta normalizada*)
  3. **Si** *estructura sintáctica es un Error* **Entonces**
  4.     *Terminar y salir*
  5. **Fin Si**
  6. *pseudo SQL*  $\leftarrow$  *analizar semánticamente* (*estructura sintáctica*)
  7. *SQL final*  $\leftarrow$  *analizar pragmáticamente* (*pseudo SQL*)
  8. *resultado*  $\leftarrow$  *consultar base de datos con* (*SQL final*)
  9. *mostrar resultado por medio del front – end*
- 

---

##### **Algoritmo 2:** Analizar morfológicamente

---

**Entrada**  $c_0$ : consulta del usuario en LN

1. *consulta*  $\leftarrow$   $c_0$
  2. *consulta*  $\leftarrow$  *convertir a minúsculas* (*consulta*)
  3. *consulta*  $\leftarrow$  *eliminar acentos* (*consulta*)
  4. *consulta*  $\leftarrow$  *eliminar puntuación superflua* (*consulta*)
  5. *consulta*  $\leftarrow$  *formatear fechas y horas* (*consulta*)
  6. *consulta*  $\leftarrow$  *reemplazar operadores "menor", "mayor", ... por <, >, ...; respectivamente* (*consulta*)
  7. *consulta*  $\leftarrow$  *eliminar espacios repetidos o agregarlos* (*consulta*)
  8. *consulta*  $\leftarrow$  *tokenizar cada elemento léxico* (*consulta*)
  9. *consulta normalizada*  $\leftarrow$  *obtener lema del diccionario* (*consulta*)
  10. **Retornar** *consulta normalizada*
- 

Sobre el algoritmo anterior cabe reseñar lo siguiente:

- El paso 5 convierte, p.ej., “11 de julio de 2015” en “11-07-2015”.
- El paso 8 separa cada token.
- La línea 9 obtiene el lema y el símbolo de la categoría gramatical de cada elemento del array (véase el Anexo 7).

Realizado el análisis morfológico se realiza el sintáctico, que opera sobre un autómata que devuelve una lista de pasos para generar una consulta SQL, o un valor centinela en caso haya error. No se incluye el algoritmo del autómata, pero su diseño se puede ver en el acápite 4.1.2.7:

---

**Algoritmo 3:** Análisis sintáctico

---

**Entrada**  $c_N$ : consulta normalizada

1. *estructura sintáctica*  $\leftarrow$  recorrido de la red de transiciones ( $c_N$ )
  2. **Retornar** *estructura sintáctica*
- 

La estructura sintáctica obtenida en el algoritmo anterior se construye por pares nombre de función/lista de parámetros, que son interpretados para generar un “pseudo SQL”:

---

**Algoritmo 4:** Analizar semánticamente

---

**Entrada** L: estructura sintáctica

1. **Para cada** elemento en lista L **Hacer**
  2.     *función*  $\leftarrow$  elemento [nombre de función]
  3.     *parámetro*  $\leftarrow$  elemento [lista de parámetros]
  4.     *agregar a comandos* (*interpretar* (*función*, *parámetro*))
  5. **Fin Para**
  6. *pseudo SQL*  $\leftarrow$  crear consulta (*comandos*)
  7. **Retornar** *pseudo SQL*
-

El algoritmo previo genera una cadena, que incluye elementos no pertenecientes al SQL estándar. El siguiente algoritmo los interpreta y, si se necesitara, obtiene combinaciones entre tablas, para devolver la consulta SQL final.

---

**Algoritmo 5:** Analizar semánticamente

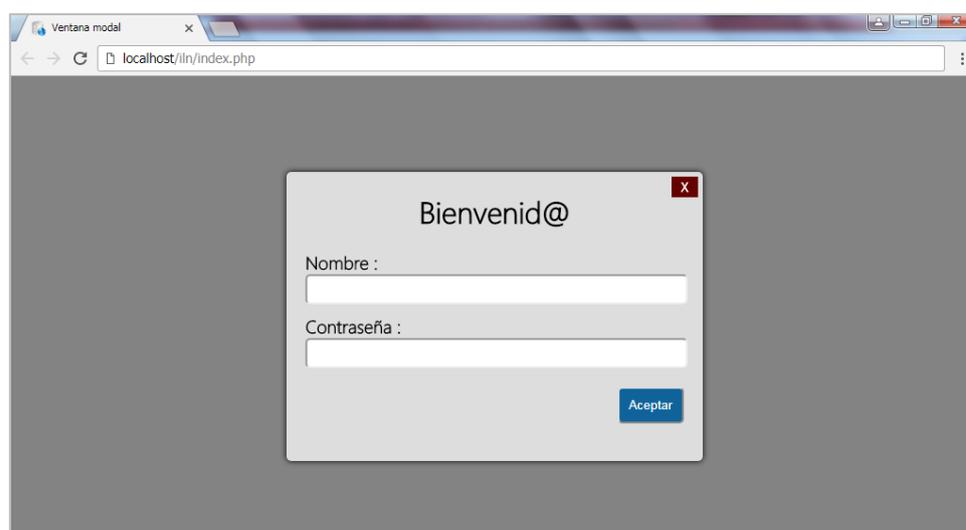
---

**Entrada S:** pseudo SQL

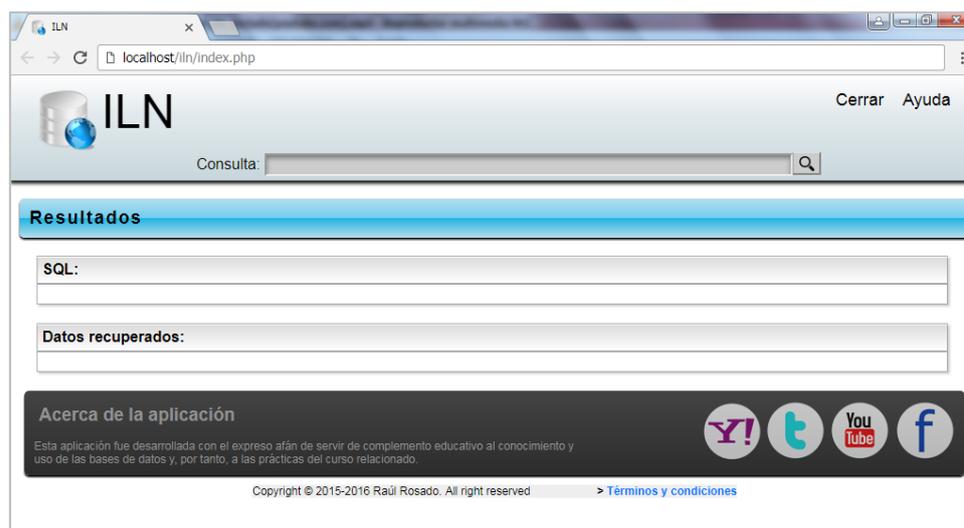
- 1: *interpretar operador de literal cadena en S*
  - 2: *interpretar operador de igualdad para cadenas en S*
  - 3: *interpretar operador de desigualdad para cadenas en S*
  - 4: *interpretar los caracteres comodines de la consulta en S*
  - 5: *obtener las combinaciones en S*
  - 6: *nuevo SQL  $\leftarrow$  crear nuevo SQL (S)*
  - 7: **Retornar** nuevo SQL
- 

#### 4.1.3.4. Resultados de la implementación

Existe la intención de simular un proceso de búsqueda corriente, ocultando la implementación y dando la ilusión de simplicidad, como se muestra a continuación (véase también el Anexo 9).



**Figura 26.** Captura del aspecto inicial del sistema de búsqueda



**Figura 27.** Captura de la pantalla principal

#### 4.1.3.5. Métrica de código fuente

Se utiliza la métrica de Halstead (Pressman, 2010), que se vale de una serie de partículas a contar: los operandos (variables o constantes que simbolizan datos), y los operadores (acciones algorítmicas que afectan el valor u orden de un operando).

Al identificar ambos se puede contar, y por tanto medir, el software con las siguientes métricas: conteo de operadores diferentes ( $\eta_1$ ), conteo de operandos distintos ( $\eta_2$ ), número total de veces que aparecen los operadores ( $N_1$ ), y el número total de veces que aparecen los operandos ( $N_2$ ).

La ILN,  $n_1$  está constituida por los elementos del lenguaje que aparecen en la aplicación y que están predefinidos por el lenguaje PHP (The PHP Group, 2016), como palabras reservadas, calificadores, operadores matemáticos, etc.; los cuales totalizan 85. Y, aparecen en la aplicación un total ( $N_1$ ) de 4400 veces.

Mientras que  $n_2$ , es decir, variables, literales y constantes definidos por el programador, son 181; apareciendo un total ( $N_2$ ) de 1370 veces.

A partir de las métricas básicas se puede definir el vocabulario del programa ( $\eta$ ) como la suma de las distintas partículas usadas, así:

$$\eta = \eta_1 + \eta_2 = 85 + 181 = 266$$

Con los valores  $N_1$  y  $N_2$  se calcula la longitud del programa ( $N$ ), en función de los operandos y operadores:

$$N = N_1 + N_2 = 4400 + 1370 = 5770$$

Este valor indica que se trata de un proyecto aceptable. Su volumen ( $V$ ):

$$V = N \times \log_2(\eta)$$

$$V = 5770 \times \log_2(266)$$

$$V \cong 46478.98$$

Se explica como el número de comparaciones mentales necesarias para escribir el programa de longitud  $N = 5770$  con su vocabulario de tamaño  $\eta = 266$ , no siendo muy complejo y, por tanto, resultando aceptable.

#### 4.1.4. Test y pruebas

##### 4.1.4.1. Fase de pruebas

Aunque avanzada la aplicación no se considera terminada, porque aún hay tareas pendientes referidas a entender mejor la necesidad

informativa del usuario, palabras inexistentes en el diccionario o errores de escritura que el analizador sintáctico decide ignorar.

Sin embargo, se han realizado pruebas unitarias y de integración en el sistema de búsqueda a la par de su implementación modular. Se muestran algunas pruebas realizadas:

**Cuadro 24.** Pruebas de acceso a la base de datos

Requerimiento	Resultado esperado	Resultado obtenido
Acceder a la base de datos	Acceder desde el módulo Datos	Correcto
Conectarse a la base de datos	Conexión a la base de datos por medio del módulo Datos	Correcto
Ejecutar consultas SQL desde PHP	Ejecución de consultas SQL por medio del módulo Datos	Correcto

**Cuadro 25.** Pruebas sobre la aplicación

Requerimiento	Resultado esperado	Resultado obtenido
Acceder a la aplicación desde el sistema	Enviar una consulta en LN, traducirla a su equivalente en SQL, y mostrar un mensaje adecuado	Correcto
Enviar consultas a la base de datos	Enviar la consulta SQL creada por la aplicación y obtener una respuesta de la base de datos	Correcto
Visualizar resultados	Visualizar las respuestas adecuadamente	Correcto

#### 4.1.4.2. Métrica de prueba

Esta métrica evalúa el esfuerzo de las pruebas, de modo que se recurre a los cálculos obtenidos por las medidas de Halstead. Así, mediante:

$$NP = \frac{1}{\left[\left(\frac{\eta_1}{2}\right) \times \left(\frac{N_2}{\eta_2}\right)\right]}$$

Que da como resultado  $0.00310863 \cong 0.003$ . Donde  $NP$  representa el nivel del programa;  $\eta_1$  el número de operadores distintos;  $\eta_2$  la cantidad de distintos operandos; y  $N_2$  el número total de veces que aparecen los operandos (valores expuestos en 4.1.3.5).

Igualmente, considerando el volumen del programa ( $V$ ), calculado en el acápite 4.1.3.5, se computa el esfuerzo ( $e$ ), que puede entenderse como la dificultad de entender e implementar el programa, es decir, la medida de su claridad:

$$e = \frac{V}{NP}$$

Resultando 14951594.42, con un aproximado de 15.49 parte del volumen del programa con capacidad de bugs. Lo que demuestra que es aceptable el esfuerzo para realizar pruebas.

**Cuadro 26.** Resumen de indicadores de la métrica de prueba

Nivel del programa	Volumen del programa	Esfuerzo (e)	Tiempo (seg.) e/18	Errores (bugs) V/3000
0.003	46,478.98	14'951,594.42	830,644.13	15.49

#### 4.2. RESULTADOS DE LA PRUEBA DE PERFORMANCE

La prueba de performance (Anexo 4) aplicada a la muestra ofrece datos sobre cómo resuelven los alumnos (filas del 1 al 18 hombres, y del 19 al 31 mujeres) las consultas planteadas, usando primero el SQL, y luego mediante la interfaz de lenguaje natural.

Notar que las diez preguntas (columnas numeradas del 1 al 10) son las mismas antes y después del experimento. Cada pregunta vale dos puntos (uno por enunciado propuesto). Al final, se suma el puntaje obtenido y se le valora de acuerdo a la Operacionalización de variables, es decir, si el alumno obtiene entre 00-05 se considera un manejo de las bases de datos “muy deficiente”, entre 06-10 “deficiente”, desde 11 hasta 13 “regular”, “bueno” si tiene entre 14-17, y “muy bueno” si tiene de 18 a 20.

**Cuadro 27.** Resultados de la prueba de performance presentada en el Anexo 4

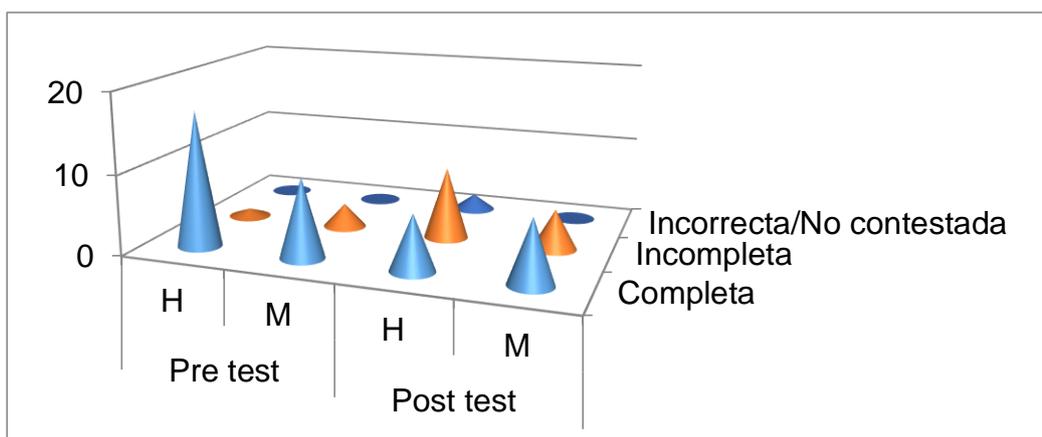
Sexo	Alumno	Pre test										Total	Post test										Total
		1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10	
Hombre	1	2	2	2	2	2	1	0	0	0	0	11	1	2	0	2	2	2	2	2	2	1	16
	2	2	2	2	1	2	1	0	1	0	0	11	2	2	2	2	2	1	2	2	1	2	18
	3	2	2	1	2	2	1	0	0	0	0	10	1	1	2	2	2	1	2	2	2	1	16
	4	2	2	2	2	2	1	1	0	0	0	12	1	2	1	2	2	2	2	2	2	2	18
	5	2	2	2	1	2	0	1	0	1	1	12	2	2	2	2	2	1	2	2	2	2	19
	6	2	2	2	2	2	1	0	0	1	0	12	1	2	2	2	2	2	1	2	1	2	17
	7	2	1	2	0	1	0	0	0	0	0	6	0	1	2	2	2	1	2	2	2	2	16
	8	2	2	1	1	2	1	0	0	0	0	10	1	2	2	2	2	2	2	2	2	2	19
	9	2	2	2	2	2	1	0	0	0	0	11	1	2	2	2	2	2	2	2	2	2	19
	10	2	2	2	2	2	1	0	0	0	0	11	1	2	2	2	2	2	2	1	2	2	18
	11	2	2	2	2	2	1	0	1	0	0	12	2	1	2	2	2	2	1	1	2	2	17
	12	2	2	2	2	2	1	0	0	0	0	11	2	2	2	2	2	1	2	2	2	2	19

Mujer	13	1	2	2	2	2	0	1	0	0	0	10	2	1	1	2	2	2	2	1	2	2	17	
	14	2	2	2	2	2	1	1	0	0	0	12	0	2	2	2	2	2	2	2	2	2	18	
	15	2	2	2	2	2	1	0	0	0	0	11	1	1	2	2	2	2	2	2	2	2	18	
	16	2	2	1	1	2	0	0	0	0	0	8	2	2	2	2	2	2	2	2	2	2	2	20
	17	2	2	1	1	2	1	0	0	0	0	9	1	2	2	2	2	2	2	2	2	2	2	19
	18	2	2	2	2	2	1	0	0	0	0	11	2	2	2	2	2	2	2	2	2	2	2	20
	19	2	2	2	1	2	1	0	0	0	0	10	1	1	2	2	2	1	2	2	2	2	2	17
	20	2	2	2	1	2	0	0	0	0	0	9	2	1	2	2	2	2	2	2	2	2	2	19
	21	2	2	2	1	2	1	0	0	0	0	10	2	2	2	2	2	2	2	2	2	2	2	20
	22	1	2	2	2	2	1	0	0	0	0	10	1	2	1	2	2	1	2	2	2	2	2	17
	23	2	2	2	2	2	1	0	0	0	0	11	2	2	2	2	2	1	2	1	1	2	2	17
	24	2	2	1	0	2	1	0	0	0	0	8	2	1	1	2	2	2	2	2	2	2	2	18
	25	2	1	1	0	2	1	1	0	1	0	9	1	2	2	2	2	2	2	2	1	2	2	18
	26	2	2	2	2	2	1	0	0	0	0	11	2	1	2	2	2	1	1	2	2	2	2	17
	27	2	2	2	0	2	0	0	0	0	0	8	2	1	2	2	2	2	2	2	1	1	1	17
	28	1	1	2	1	2	1	0	0	0	0	8	1	2	2	2	2	2	2	1	2	2	2	18
	29	1	2	2	2	1	1	0	0	0	0	9	2	2	2	2	2	2	1	2	2	1	2	18
	30	2	2	2	1	2	0	0	0	0	0	9	1	2	1	2	2	2	2	1	2	2	2	17
31	2	2	1	2	2	0	1	0	0	0	10	2	2	2	2	2	2	2	2	2	2	2	20	

4.2.1. Respecto del primer enunciado de la prueba de performance

**Cuadro 28.** Frecuencias de las respuestas del primer enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	17	10	27	7	8	15
Incompleta	1	3	4	9	5	14
Incorrecta/No contestada	0	0	0	2	0	2
Total	18	13	31	18	13	31



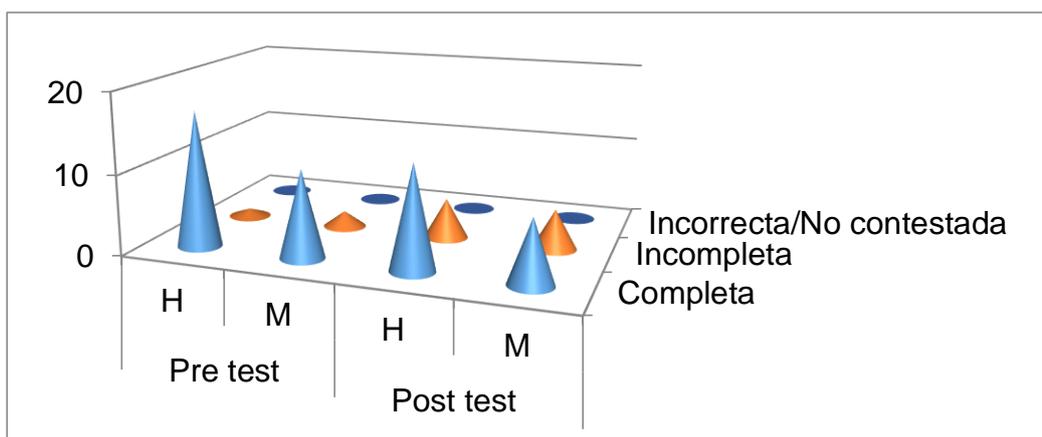
**Figura 28.** Frecuencias de las respuestas del primer enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Los datos anteriores muestran una clara diferencia a favor del pre test. Las consultas de una sola tabla suelen ser básicas en SQL y no requieren mayor adiestramiento. En el caso del software experimental se comprueban ciertas desventajas (ver el Cuadro 2), como el que los usuarios creen que la ILN hace cosas que en realidad no puede, como bien anticipa Cócera (2011), por causa del dominio lingüístico para elaborar consultas.

**4.2.2. Respecto del segundo enunciado de la prueba de performance**

**Cuadro 29.** Frecuencias de las respuestas del segundo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	17	11	28	13	8	21
Incompleta	1	2	3	5	5	10
Incorrecta/No contestada	0	0	0	0	0	0
Total	18	13	31	18	13	31



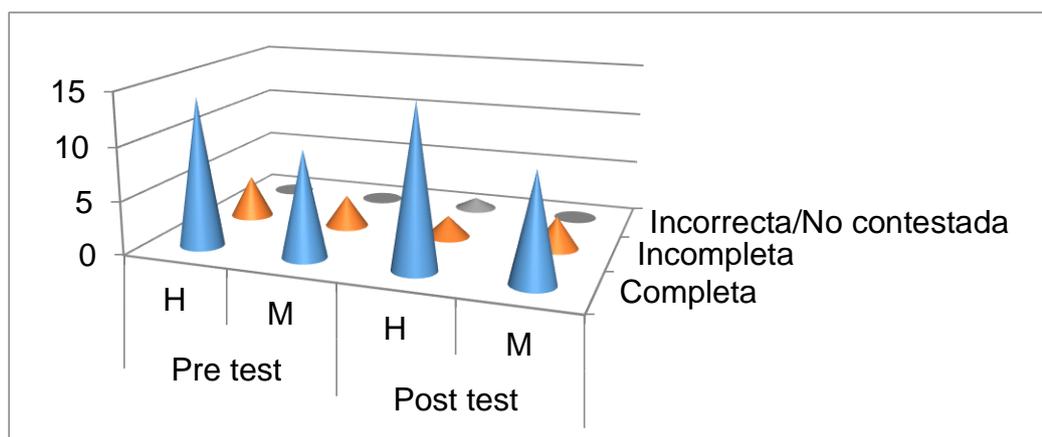
**Figura 29.** Frecuencias de las respuestas del segundo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

El cuadro y gráfico previos, exponen algún acercamiento entre los resultados pre y post test. Las consultas siguen siendo básicas, de manera que el SQL requiere aún poco adiestramiento. Pero se advierte que una consulta, cuanto más específica, más fácil es de plantear con una ILN. En consecuencia, el lenguaje natural resultará más cómodo que uno formal, reafirmando así a J. J. González, et ál. (2012), en el sentido de que una ILN facilitan la interacción hombre-máquina.

**4.2.3. Respecto del tercer enunciado de la prueba de performance**

**Cuadro 30.** Frecuencias de las respuestas del tercer enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	14	10	24	15	10	25
Incompleta	4	3	7	2	3	5
Incorrecta/No contestada	0	0	0	1	0	1
Total	18	13	31	18	13	31



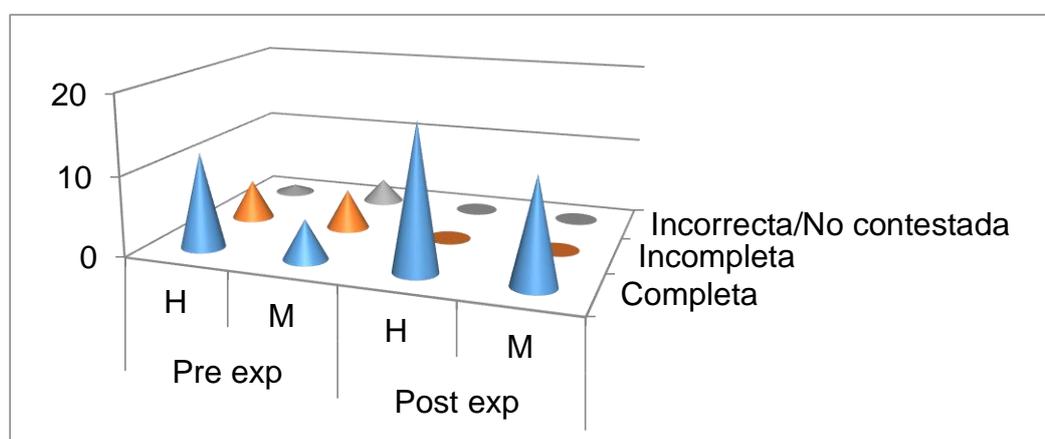
**Figura 30.** Frecuencias de las respuestas del tercer enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

El cuadro y gráfico preliminares muestra la ventaja del post test sobre su contraparte. Tal ventaja radica en que las consultas prácticamente se transcriben desde su enunciado por ser, parafraseando a Russell y Norvig (2010), la forma de comunicación natural entre personas. Un lenguaje formal puede producir errores lógicos, si se desconoce la precedencia entre operadores. Por el contrario la ILN capta el sentido de la consulta imitando (o, al menos, pareciendo imitar), en este aspecto, la inteligencia humana, como dice López-Cózar (2013). Lo que la convierte en una aplicación PLN, según entiende Gelbuk (2010). Sobre la comprensión del idioma que parece tener la ILN, ésta modela sólo una parte de aquél (Moreno, 2009). A esto se le llama “el dominio del idioma”, que está restringido a la propia base de datos como remarcan Aguirre y Pazos (2012), y Montalbán y Castro (2014).

4.2.4. Respetto del cuarto enunciado de la prueba de performance

**Cuadro 31.** Frecuencias de las respuestas del cuarto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	12	5	17	18	13	31
Incompleta	5	5	10	0	0	0
Incorrecta/No contestada	1	3	4	0	0	0
Total	18	13	31	18	13	31



**Figura 31.** Frecuencias de las respuestas del cuarto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

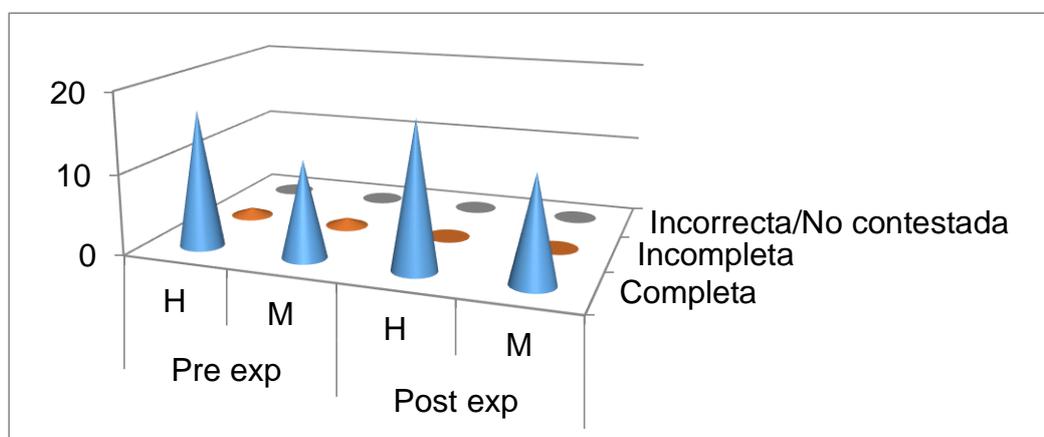
El post test sobrepasa al pre test en este resultado, debido a que el lenguaje natural no exige detalles de la base de datos. Si bien el usuario de la ILN necesita conocer el nombre de las tablas y los campos que guarda, no requiere conocer todo el esquema exacto de la base de datos, ni las relaciones entre sus tablas. Esto gracias a uno de los procesos de los que hablan J. J. González et ál. (2012): “el análisis del discurso”, y que logra diferenciar una aplicación PLN de otra que no lo es (Gelbukh, 2010).

Por otra parte, destaca la importancia de una gramática para evaluar si las oraciones escritas son correctas (Pereiras, 2011), ya que su aceptación depende de un autómata el que ayuda a formar oraciones según Chomsky (Barón & Müller, 2014).

**4.2.5. Respecto del quinto enunciado de la prueba de performance**

**Cuadro 32.** Frecuencias de las respuestas del quinto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	17	12	29	18	13	31
Incompleta	1	1	2	0	0	0
Incorrecta/No contestada	0	0	0	0	0	0
Total	18	13	31	18	13	31



**Figura 32.** Frecuencias de las respuestas del quinto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

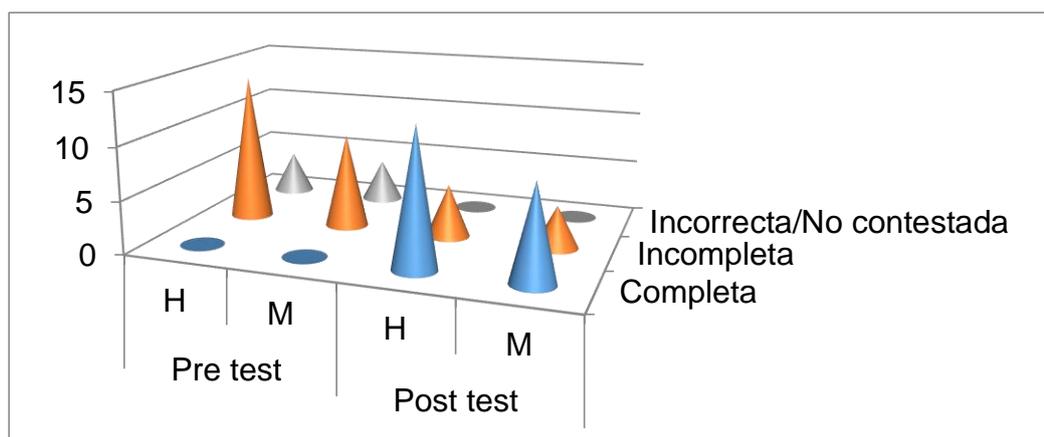
En este punto se muestra cierta ventaja del post sobre el pre test. Esto se debe a la sencillez de uso de la ILN, en contraste con el SQL, que es un lenguaje de programación para bases de datos (Silberschatz, Korth,

& Sudarshan, 2011). Este requiere conocer sus reglas y mucha práctica. Ahora bien, este tipo de práctica debe lograr en el aprendiz un tipo de conocimiento asimilable para aplicarse en otros ambientes, y no sólo mecánico, como bien afirma (Sánchez, s.f.), lo que lleva tiempo. Mientras que la ILN, parece entender al usuario (López-Cózar, 2013), sin que éste necesite nada más que una redacción apropiada.

**4.2.6. Respecto del sexto enunciado de la prueba de performance**

**Cuadro 33.** Frecuencias de las respuestas del sexto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	0	0	0	13	9	22
Incompleta	14	9	23	5	4	9
Incorrecta/No contestada	4	4	8	0	0	0
Total	18	13	31	18	13	31



**Figura 33.** Frecuencias de las respuestas del sexto enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

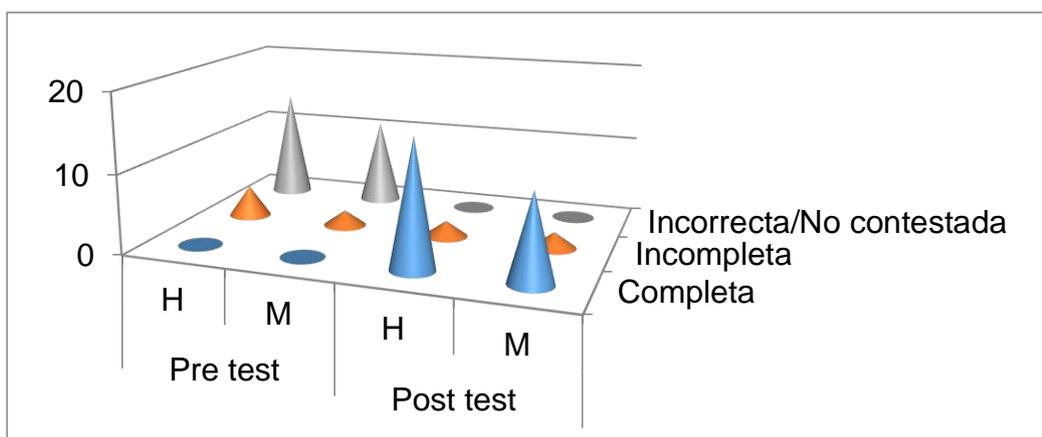
El gráfico anterior y el cuadro del que deriva presentan enormes diferencias entre las etapas pre y post experimental. El SQL, aun teniendo características que lo acercan al lenguaje humano (Cócera, 2011), no puede interrogar a una base de datos con palabras clave.

El lenguaje natural, bien redactado, tampoco permite las palabras clave; pero muchas expresiones tienen un carácter lacónico. La interfaz intenta emular tal característica y por ello aventaja al SQL, ya que facilita la comunicación humano-computadora (obviando tener que escribir mucho texto para pedir información muy puntual). Esta cualidad es bien recibida por los usuarios, por ser propia de los motores de búsqueda actuales a los cuales están adaptados, lo que termina por influir en el uso de las bases de datos (Kendall & Kendall, 2011), y en la creación de ILNBDs como se deduce de Shabaz et ál. (2015).

#### 4.2.7. Respecto del séptimo enunciado de la prueba de performance

**Cuadro 34.** Frecuencias de las respuestas del séptimo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	0	0	0	16	11	27
Incompleta	4	2	6	2	2	4
Incorrecta/No contestada	14	11	25	0	0	0
Total	18	13	31	18	13	31



**Figura 34.** Frecuencias de las respuestas del séptimo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

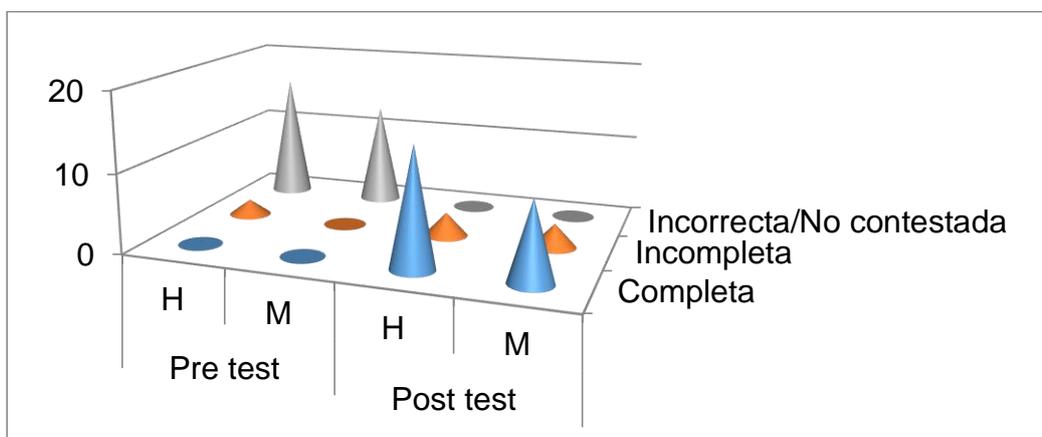
El cuadro y figura previos muestran como el lenguaje natural es capaz de relacionar conceptos e ideas de manera implícita, pero realizando análisis exhaustivos con el fin de comprender la oración completamente como dicen Zamoszczyk et ál. (2012). En los enunciados propuestos el SQL, necesita que el usuario eche mano de lo que se conoce como “combinaciones”. Muchos usuarios informáticos principiantes desconocen este tema, por el nivel técnico que demanda. Aun así, la popularidad de las bases de datos (Ricardo, 2009) hace necesario conocer dicho tema para saber relacionar tablas apropiadamente.

#### 4.2.8. Respecto del octavo enunciado de la prueba de performance

**Cuadro 35.** Frecuencias de las respuestas del octavo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	0	0	0	15	10	25
Incompleta	2	0	2	3	3	6

Incorrecta/No contestada	16	13	29	0	0	0
Total	18	13	31	18	13	31



**Figura 35.** Frecuencias de las respuestas del octavo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Nuevamente se aprecian las ventajas del lenguaje natural, puesto que este tipo de consultas son más sencillas de enunciar con el lenguaje natural que con SQL, a pesar de que la interpretación del lenguaje no es tarea sencilla como aclara Alconada (2013).

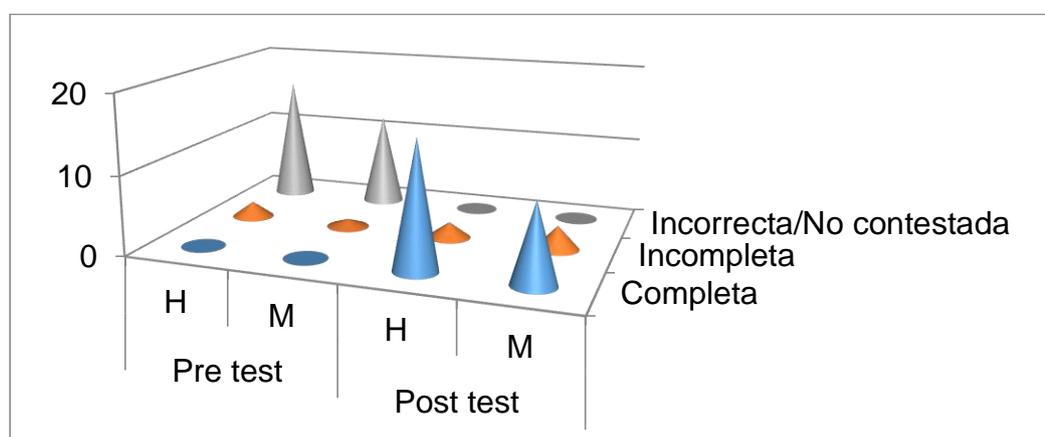
Más allá de la discusión del apartado precedente, lo cierto es que relacionar diferentes tablas puede conllevar problemas y errores, como p.ej. al existir campos con el mismo nombre en diferentes tablas, ya que la computadora no sabrá a cuál de ellos se hace referencia en un momento determinado.

Este tipo de problemas son solucionados por el PLN y liberan al usuario de la ILN de esos detalles.

4.2.9. Respeto del noveno enunciado de la prueba de performance

**Cuadro 36.** Frecuencias de las respuestas del noveno enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	0	0	0	16	10	26
Incompleta	2	1	3	2	3	5
Incorrecta/No contestada	16	12	28	0	0	0
Total	18	13	31	18	13	31



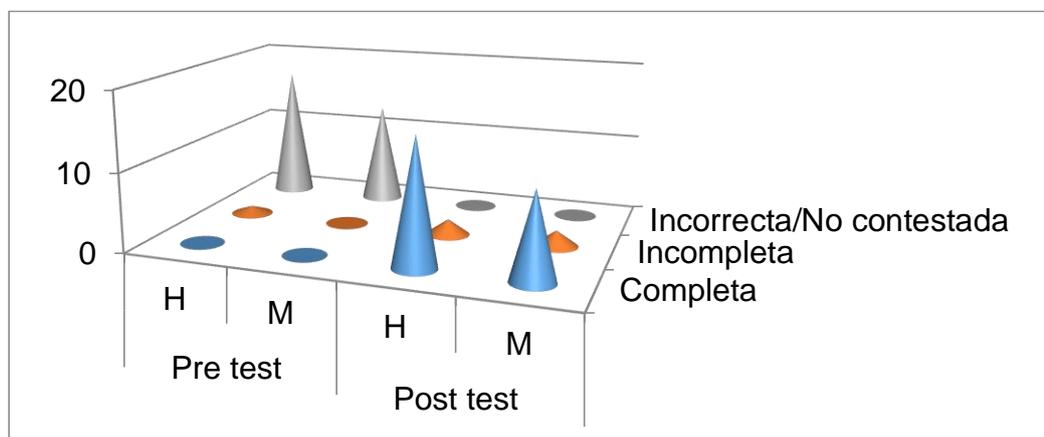
**Figura 36.** Frecuencias de las respuestas del noveno enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

De todos los tipos de consulta, éste bien podría ser considerado el más complejo. No sólo hay que saber cómo relacionar más de una tabla, sino que también hay que proyectar correctamente las relaciones mediante SQL. La ILN sorteó esto con sólo copiar el enunciado, como descubrieron por sí mismos los usuarios. De modo que se ratifica la idea expuesta por Rodríguez (2003), de que el idioma puede controlar una aplicación software.

4.2.10. Respeto del décimo enunciado de la prueba de performance

**Cuadro 37.** Frecuencias de las respuestas del décimo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

Número de respuestas	Pre test			Post test		
	H	M	Total	H	M	Total
Completa	0	0	0	16	11	27
Incompleta	1	0	1	2	2	4
Incorrecta/No contestada	17	13	30	0	0	0
Total	18	13	31	18	13	31



**Figura 37.** Frecuencias de las respuestas del décimo enunciado de la prueba de performance. Estudiantes del V semestre de la carrera de Computación e Informática, IESTP Juli, 2016

En este punto se aprecia como la utilización de la ILN supera en gran medida al uso del SQL. En este caso las consultas, cuya engañosa sencillez contrasta con el hecho de estar expresamente diseñadas para demostrar lo laborioso de su planteamiento con SQL, prueban la conveniencia del uso del lenguaje natural con respecto a otro formal. Hay que notar que, al final del día, una aplicación de base de datos será siempre deseable; pero que un mecanismo que facilite dicho acceso

será tanto o más necesario para que gente común pueda acceder a la información que guarda, como afirman acertadamente Aguirre y Pazos (2012), y Bender et ál. (2014). P.ej., la ILN, mediante un recorrido de grafos (Goodrich, Tamassia, & Goldswasser, 2014), intenta simplificar el proceso de relacionar dicha información que, de otra forma, obligaría a los usuarios a hacerlo por ellos mismos.

### 4.3. PRUEBA DE HIPÓTESIS

De acuerdo al punto 3.8 (Diseño estadístico para la prueba de hipótesis) se siguen los siguientes pasos:

1) Planteamiento de hipótesis:

$H_0: \mu_A = \mu_B$  La valoración general del nivel de uso procedimental de las bases de datos pre test es igual a la valoración general del nivel de uso procedimental post test, es decir, no mejora en absoluto (Hipótesis nula).

$H_1: \mu_A \neq \mu_B$  La valoración general del nivel de uso procedimental de las bases de datos pre test es diferente a la valoración general del nivel de uso procedimental post test (Hipótesis alterna).

- 2) El nivel de significancia  $\alpha = 0.05$  o un 95% de nivel de confianza.
- 3) La regla de decisión es: si  $Z_c > Z_t$ , se rechaza  $H_0$  (hipótesis nula) y se acepta  $H_1$  (hipótesis alterna).
- 4) Para calcular  $Z_c$ , previamente, se hallan para el pre test y post test sus respectivos promedios (según la fórmula de 3.5.2.1), y varianzas (de

acuerdo a 3.5.2.2), donde  $n$  representa el tamaño del grupo experimental.

Obteniendo los siguientes resultados:

**Cuadro 38.** Resumen de datos para el cálculo de  $Z_c$

	Pre test (A)	Post test (B)
N	31	31
Promedio	10.064516	17.9677
Varianza	2.1956989	1.43226

Ahora, por la fórmula de 3.5.2.3, se tiene que:

$$Z_c = \frac{|10.064516 - 17.9677|}{\sqrt{\frac{2.1956989}{31} + \frac{1.43226}{31}}} = 16.7853045$$

- 5) Decisión: como  $Z_c = 16.7853045 > Z_t = 1.96$ , se rechaza  $H_0$ , y se acepta  $H_1$ . Entonces, el nivel de uso procedimental post test mejora (aplicando la ILN) con respecto al pre test, para un nivel de significancia de 0.05 o un nivel de confianza del 95%.

## CONCLUSIONES

- La aplicación de la ILN simplifica comprender las consultas de recuperación de información, debido a que una consulta en lenguaje humano se transforma automáticamente a su equivalente en lenguaje formal. Esto supone, a posteriori, un mejoramiento significativo en el aprendizaje de base de datos, ya que los usuarios de la ILN (probables practicantes y, por ende, potenciales usuarios del Sistema de Registro) tienen la posibilidad de familiarizarse, experimentar y conocer la manera de formular consultas apropiadas en SQL. Hecho que queda confirmado con los resultados de la sección 4.2 y la Prueba de hipótesis de la sección 4.3.
- El nivel de análisis para resolver un problema crece en proporción a la complejidad que tenga éste. Una adecuada captación de los requisitos (es decir, lo que debe hacer el sistema) permite arribar a un subsecuente buen diseño. Particularmente, el análisis orientado a objetos logra percibir de manera apropiada los requisitos de software y de usuario, como lo demuestra la sección 4.1.1.5. Por otra parte, en cuanto al software, se determinó lo adecuado de éste (sección 4.1.1.10). Como resultado de todo lo anterior, el nivel de análisis en conjunto resultó aceptable.

- El diseño, desde la óptica orientada a objetos, permite desarrollar un método para (saber cómo) solucionar un problema previamente tratado en base a un análisis hecho con una mirada, también, orientada a objetos. Este diseño evita desarrollar sistemas mal planeados, y tolera una rápida y concienzuda implementación ulterior mediante un modelo arquitectónico aceptable, como se aprecia en la métrica de diseño (sección 4.1.2.8).
- La implementación o programación orientada a objetos conlleva sencillez y naturalidad, y convierte un diseño orientado a objetos en un sistema funcional. Hoy en día la orientación a objetos es clave en la industria del software, ya que proporciona beneficios como la reutilización de código, que ahorra tiempo de desarrollo y mejora la calidad del software. Hecho que queda palpable en la sección 4.1.3.5 (métrica del código fuente). Asimismo, el software producido es más comprensible, mejor organizado y resulta más fácil de mantener, modificar y depurar, según consta en la sección 4.1.4.2, donde se detalla el trabajo para probar la ILN. Llegando con todo lo anterior, como resultado, a un nivel aceptable del esfuerzo de implementación y testeado del software.
- Los resultados expuestos en el Cuadro 38 de la sección 4.3, muestran que el promedio del uso de las bases de datos relacionales pasa de 10.064516 (nivel “deficiente”, según la Operacionalización de variables) a un contundente 17.9677 (nivel “bueno”), confirmando así que la ILN mejora el nivel procedimental de los usuarios, especialmente donde las correspondencias entre datos no son tan obvias o de plano son tácitas, y es difícil enunciar en SQL lo que se busca y cómo obtenerlo.

## RECOMENDACIONES

- Se debe de lograr aplicar la ILN a la mayor gama de bases de datos posible (de tipo no relacional y/o con manejadores distintos de MySQL). Para ello, se debería hacer que el grafo, que establece la conexión entre tablas, se adapte dinámicamente a cualquier base de datos. Se recomienda aplicar el lenguaje natural en otros escenarios como en la búsqueda por Internet o cualquier tipo de repositorio, para que una persona común pueda acceder con simplicidad y precisión a cualquier acervo informativo.
- Se debe incidir en el análisis de requisitos ya que, sus ventajas durante el ciclo de vida del software, compensan ampliamente su demora. Si bien el análisis orientado a objetos es tan poderoso que permite la detección tanto de sus clases participantes como de la mutua interacción entre ellas; si no se sabe a ciencia cierta lo que desea el cliente, se puede terminar solucionando un problema totalmente distinto del que se tenía que resolver en realidad.

- Se recomienda el diseño orientado a objetos para realizar arquitecturas de software cuya complejidad es muy alta, y que involucran exigencias como patrones de diseño, reutilización y refactorización (para una proporcionar una solución mejores); características que, simplemente, son parcial o totalmente inaplicables en otros paradigmas.
- Además de las consultas de recuperación, se podrían implementar otros tipos para insertar, eliminar y modificar datos.
- Se podría incluir un módulo de voz que simplifique la interacción hombre/máquina para beneficiar, especialmente, a usuarios discapacitados. Igualmente, sería preciso contar con un módulo como el *FreeLing*, que permite un mejor tratamiento morfosintáctico. Mejorar el análisis del lenguaje mediante grafos semánticos. Y por último, podría implementarse el PLN en el aprendizaje de la programación de computadoras en general, escribiendo un pseudocódigo y traduciéndolo a algún lenguaje informático.

## BIBLIOGRAFÍA

- Aguirre, M. A., & Pazos, R. A. (22 de Octubre de 2012). Interfaces de lenguaje natural para consultar bases de datos en español. (L. Cruz Reyes, Ed.) *Komputer Sapiens*, *II*(4), 20-24.
- Alconada, D. (2013). *AlcoNQL Herramienta de consulta SQL por medio de lenguaje natural*. Universisda Oberta de Catalunya, Ingeniería Informática. Barcelona: Universitat Oberta de Catalunya.
- Barón, L., & Müller, O. (14 de Mayo de 2014). La teoría lingüística de Noam Chomsky: del inicio a la actualidad. *Lenguaje*, *42*(2), 417-422.
- Bender, C. M., Deco, C., González, J. S., Hallo, M., & Ponce, J. C. (2014). *Tópicos avanzados de bases de datos* (Primera ed.). Iniciativa Latinoamericana de Libros de Texto Abiertos.
- Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., & Houston, K. A. (2007). *Object-Oriented Analysis and Design with Applications* (Tercera ed.). Boston, MA, United States of America: Pearson Education, Inc.

- Camps, R., Casillas, L. A., Costal, D., Gibert, M., Martín, C., & Pérez, O. (2010). *Bases de datos* (Primera ed.). Barcelona, Cataluña, España: Fundació per la Universitat Oberta de Catalunya.
- Cócera, C. (2011). *Diseño y codificación de un complemento multilinguaje integrado en paquetes de productividad para facilitar la transformación del lenguaje natural en sentencias de interacción con bases de datos*. Proyecto de fin de carrera, Universidad Carlos III de Madrid, Ingeniería Técnica en Informática de Gestión, Madrid.
- Gelbukh, A. (2010). Procesamiento de lenguaje natural y sus aplicaciones. *Komputer Sapiens*, *II*(2), 6-11.
- González, J. J., Morales, M. L., & Florencia, R. (22 de Octubre de 2012). Interfaz de lenguaje natural bajo un enfoque semántico. (L. Cruz Reyes, Ed.) *Komputer Sapiens*, *I*(4), 25-27.
- Goodrich, M. T., Tamassia, R., & Goldswasser, M. H. (2014). *Data structures & algorithms in Java* (Sexta ed.). (B. Lang Golub, Ed.) Danvers, MA, United States of America: John Wiley & Sons, Inc.
- Hernández, R., Fernández, C., & Lucio, P. (2010). *Metodología de la investigación* (Quinta ed.). México D.F., México, México: McGraw-Hill/Interamericana Editores, S.A. de C.V.
- Johnson, R., & Kuby, P. (2012). *Estadística elemental: lo esencial* (Décima ed.). (F. d. Castro Pérez, Ed., & J. H. Romo Muñoz, Trad.) México D.F., México, México: Cengage Learning Editores, S.A. de C.V.

- Kendall, K. E., & Kendall, J. E. (2011). *Análisis y diseño de sistemas* (Octava ed.). (L. M. Cruz Castillo, Ed.) Naucalpan de Juárez, México, México: Pearson Educación de México, S.A. de C.V.
- Levine, D. M., Krehbiel, T. C., & Berenson, M. L. (2012). *Estadística descriptiva* (Primera ed.). (L. M. Cruz Castillo, Ed.) Naucalpan de Juárez, Edo. de México, México: Pearson Educación de México.
- Marqués, M. (2011). *Bases de datos* (Primera ed.). Castelló de la Plana: Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions.
- Montalbán, I., & Castro, M. d. (2014). *Gestión de bases de datos* (Segunda ed.). Madrid, Madrid, España: Ibergarceta Publicaciones, S.L.
- Pereiras, E. (2011). *LNE2NEXI: Traductor de consultas de lenguaje natural a NEXI con analizador sintáctico*. Trabajo de diploma, Universidad Central "Marta Abreu" de Las Villas, Licenciatura en Ciencia de la Computación, Santa Clara.
- Pressman, R. S. (2010). *Ingeniería del software, un enfoque práctico* (Séptima ed.). (V. C. Olguín, & J. E. Brito, Trads.) Colonia Desarrollo Santa Fe, México D. F., México: McGRAW-Hill Interamericana Editores, S.A. de C.V.
- Ricardo, C. M. (2009). *Bases de datos* (Primera ed.). (V. Campos Olguín, & J. Enríquez Brito, Trads.) Álvaro Obregón, México D.F., México: McGraw-Hill Interamericana Editores, S.A. de C.V.

- Rodríguez, H. (2003). Las interfaces en lenguaje natural. En M. A. Martí, J. A. Alonso, T. Bardia, J. Campàs, X. Gómez, J. Gonzalo, y otros, & M. A. Martí Antonín (Ed.), *Tecnologías del lenguaje* (Primera ed., págs. 130-156). Barcelona, Catalunya, España: Editorial UOC.
- Rosado, R. (2016). *Aplicación de una interfaz de lenguaje natural para la gestión administrativa del IESTP Juli*. Tesis, Universidad Nacional del Altiplano, Facultad de ingeniería mecánica eléctrica, electrónica y sistemas, Puno.
- Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence A Modern Approach* (Tercera ed.). (M. Hirsch, Ed.) Upper Saddle River, NJ, United States of America: Pearson Education, Inc.
- Shabaz, K., O'Shea, J. D., Crockett, K. A., & Latham, A. (29 de Julio de 2015). Aneesah: A conversational natural language interface to databases. *Proceedings of the World Congress on Engineering 2015, I*, págs. 227-232. London.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). *Database system concepts* (Sexta ed.). New York, NY, United States of America: The McGraw-Hill Companies, Inc.
- Zamoszczyk, C., De Luca, S., Ruiz, S., & Iturbide, L. (Diciembre de 2012). Human query language. *Ciencia y Tecnología*, 37-50.

#### REFERENCIAS WEB

- López-Cózar, R. (3 de Junio de 2013). *Ramón López-Cózar: Sistemas de diálogo basados en procesamiento del habla y multimodales*.

Recuperado el 18 de Marzo de 2014, de sitio web de Universidad de Granada: [http://www.ugr.es/~rlopezc/sistemas\\_dialogo.htm](http://www.ugr.es/~rlopezc/sistemas_dialogo.htm)

Moreno, M. L. (13 de Enero de 2009). *Capítulo 1 Interfaz general de lenguaje natural: Sistemas operativos inteligentes*. Recuperado el 1 de Mayo de 2017, de sitio web de mailxmail: [www.mailxmail.com/curso-sistemas-operativos-inteligentes/interfaz-general-lenguaje-natural](http://www.mailxmail.com/curso-sistemas-operativos-inteligentes/interfaz-general-lenguaje-natural)

Pazos, R. (23 de Setiembre de 2010). *Interfaces de lenguaje natural para consultas a bases de datos*. Recuperado el 31 de Enero de 2014, de Sitio web de Cinvestav: [http://www.tamps.cinvestav.mx/seminario\\_2010\\_11](http://www.tamps.cinvestav.mx/seminario_2010_11)

Sánchez, S. G. (s.f.). *seduca: UAEMex*. Recuperado el 07 de Mayo de 2017, de sitio web de la Universidad Autónoma del Estado de México: [http://www.seduca2.uaemex.mx/ckfinder/uploads/files/los\\_contenidos\\_de\\_ap\\_-1-\\_.pdf](http://www.seduca2.uaemex.mx/ckfinder/uploads/files/los_contenidos_de_ap_-1-_.pdf)

The PHP Group. (2016). *Referencias del lenguaje: Manual de PHP*. (S. Bakken, G. Hojtsy, H. Holzgraefe, P. Olson, & E. Schmid, Editores) Recuperado el 10 de Noviembre de 2016, de sitio web del Manual de PHP: <http://php.net/manual/es/index.php>



**ANEXOS**

**ANEXO 1. GLOSARIO DE ACRÓNIMOS DE INFORMÁTICA**

<b>AOO</b>	Análisis Orientado a Objetos
<b>BD</b>	Base de datos
<b>CI</b>	Carrera de Computación e Informática del IESTP Juli
<b>CSS3</b>	<i>Cascading Style Sheets</i> versión 3
<b>DOO</b>	Diseño Orientado a Objetos
<b>E/R</b>	Modelo de bases de datos Entidad/Relación
<b>HTML5</b>	<i>HyperText Markup Language</i> versión 5
<b>IA</b>	Inteligencia Artificial
<b>IESTP Juli</b>	Instituto de Educación Superior Tecnológico Público “Juli”
<b>IFPUG</b>	<i>International Function Point Users Group</i>
<b>IHM</b>	Interfaz Hombre-Máquina
<b>ILN</b>	Interfaz de Lenguaje Natural
<b>IPO</b>	Interacción Persona-Ordenador
<b>IU</b>	Interfaz de Usuario
<b>LF</b>	Lenguaje Formal
<b>LH</b>	Lenguaje Humano (igual que LN)
<b>LN</b>	Lenguaje Natural (iguale que LH)
<b>NEXI</b>	<i>Narrowed Extended XPath I</i>
<b>PFA</b>	Puntos de Función Ajustados
<b>PFSA</b>	Puntos de Función Sin Ajuste
<b>PHP</b>	<i>PHP Hypertext Pre-processor</i>
<b>PLN</b>	Procesamiento de Lenguaje Natural
<b>POO</b>	Programación Orientada a Objetos
<b>SQL</b>	<i>Structured Query Language</i>
<b>UML</b>	<i>Unified Modeling Language</i>

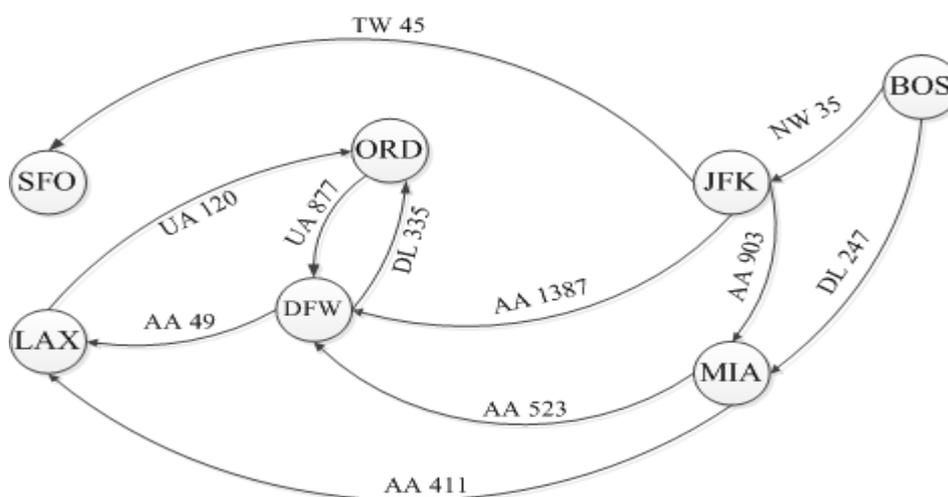
**ANEXO 2. BREVE RESEÑA SOBRE REPRESENTACIÓN Y RECORRIDO DE GRAFOS**

**Matriz de adyacencias**

Es la representación como matriz de un grafo  $G$ , cuyos vértices pertenece al conjunto  $\{1, \dots, n\}$ . La matriz  $G$  resultante se determina como sigue:

Un vértice  $v$  tendrá una clave distinta del intervalo  $1, \dots, n$ . Esto permite simplificar la descripción de los vértices, de modo que el vértice  $i$  se llamará "vértice  $i$ ".

Se mantiene un arreglo  $G$  bidimensional  $n \times n$  tal que la celda  $G[i, j]$  guarda una referencia a la arista  $e$  que conecta los vértices  $i$  y  $j$ , si existe dicha arista. Si la arista  $e$  es no dirigida, se guarda la referencia a  $e$  tanto en  $G[i, j]$  como en  $G[j, i]$ . Si no hay arista del vértice  $i$  al  $j$ ,  $G[i, j]$  refiere un objeto nulo (o algún indicador que diga que esta celda no se asocia con ninguna arista). Un ejemplo sencillo se puede ver en seguida:



**Figura 38.** Grafo dirigido

Fuente: Goodrich, Tamassia, & Goldwasser (2014, pág. 614)

**Cuadro 39.** Enumeración de los vértices del grafo anterior

1	2	3	4	5	6	7
BOS	DFW	JFK	LAX	MIA	ORD	SFO

Fuente: Figura 38

**Cuadro 40.** Matriz de adyacencias del grafo dirigido

	1	2	3	4	5	6	7
1	∅	∅	NW 35	∅	DL 247	∅	∅
2	∅	∅	∅	AA 49	∅	DL 335	∅
3	∅	AA 1387	∅	∅	AA 903	∅	TW 45
4	∅	∅	∅	∅	∅	UA 120	∅

5	∅	AA 533	∅	AA 411	∅	∅	∅
6	∅	UA 877	∅	∅	∅	∅	∅
7	∅	∅	∅	∅	∅	∅	∅

Fuente: Figura 38

### Recorrido de grafos

Es un procedimiento sistemático para explorar un grafo, examinando todos sus vértices y aristas. Es eficiente si visita todos los vértices y las aristas en un tiempo proporcional a su cantidad.

### Primera búsqueda en profundidad

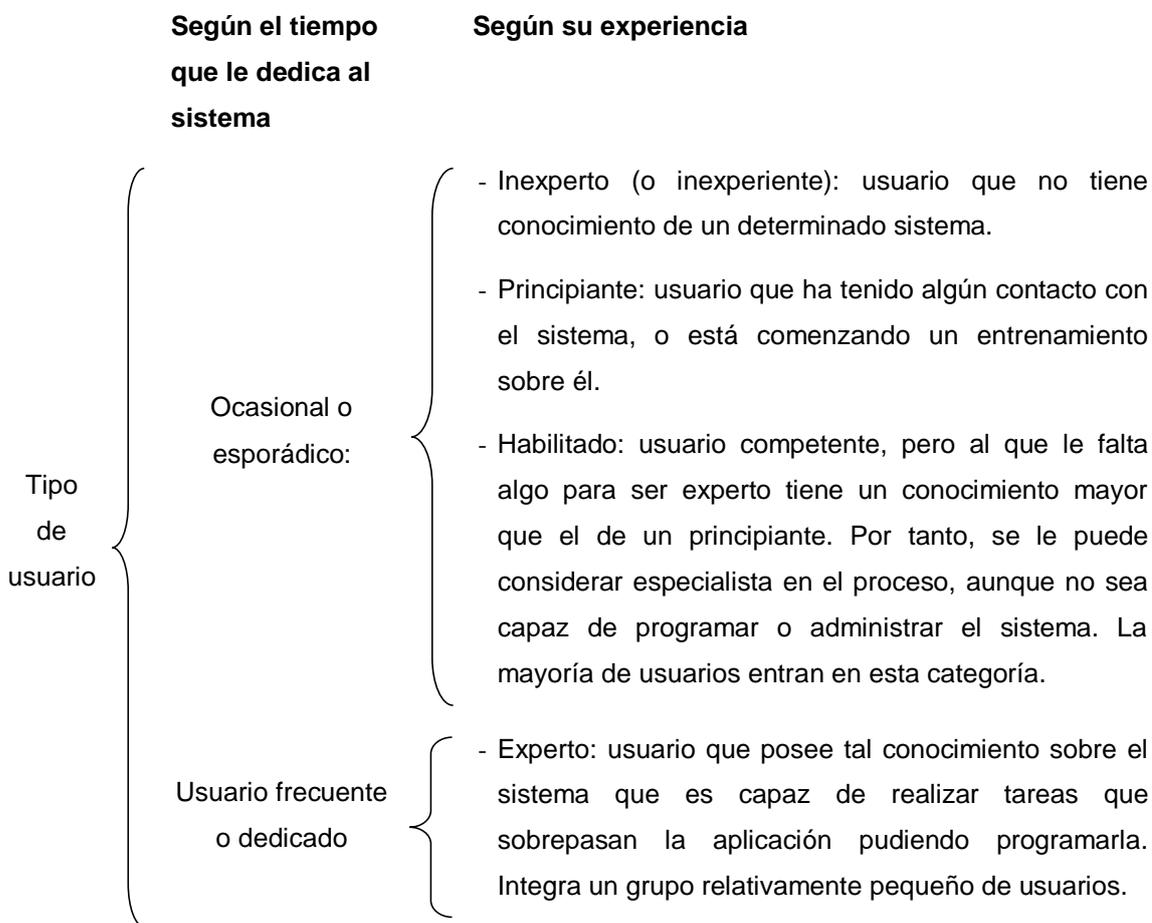
Se parece a caminar en un laberinto con un hilo y un balde de pintura para no perderse. Se comienza en un vértice específico inicial  $s$  en  $G$ , que se inicializa fijando un extremo del hilo a  $s$ , y se pinta como “visitado”, siendo ahora el vértice “actual”, llamado  $u$ . A continuación se recorre  $G$  considerando una arista arbitraria  $(u, v)$ . Si  $v$  ya fue visitado, de inmediato se regresa a  $u$ ; si no, se desenrolla el hilo y se pasa a  $v$  que será el nuevo vértice actual pintándose como “visitado” y repitiéndose el procedimiento anterior. Si se llega a un vértice actual  $u$  tal que todas las aristas convergen en vértices ya visitados, se enrolla el hilo, retrocediendo por la arista y pasando al vértice  $v$  que condujo a  $u$ . Entonces se hace que  $v$  sea el vértice actual y se repiten los procesos previos para una arista que todavía no se haya visitado. Se continúa así, avanzando y retrocediendo hasta que se encuentre un vértice con trayectorias todavía inexploradas. El proceso termina cuando el retroceso conduce de nuevo a  $s$  y no hay más aristas inexploradas que incidan en  $s$ .

**ANEXO 3. USUARIO INFORMÁTICO**

**¿Qué es un usuario informático?**

Usuario informático es un conjunto de permisos y dispositivos a los cuales puede tener acceso una persona (máquina o software) para interactuar con un programa informático, ya sea como operador (usuario directo), o como una persona (máquina o software) beneficiaria de algún modo con las operaciones del sistema (usuario indirecto). El usuario u operador directo se conoce también como usuario final, es decir, aquel usuario al cual se le dedica el programa informático y que, en última instancia, lo usará en sus labores corrientes (ya sea que lo haya adquirido o no).

Cuando se desarrollan Interfaces Hombre-Máquina (IHM) se debe empezar por conocer a los potenciales usuarios por medio del *modelo de usuario* (el perfil de los usuarios finales del sistema: edad, sexo, estudios, capacidades físicas, personalidad, etc.) En este estudio, para fines de evaluación, destaca la *experiencia del usuario* (el nivel de conocimiento del usuario sobre el sistema: avanzado, principiante, intermedio), así como el tiempo que los usuarios interactúan con el sistema. Pudiendo categorizárseles de esta manera:



#### **ANEXO 4. PRUEBA DE PERFORMANCE**

**(Los tipos de consultas se realizaron en base al Anexo 10. Clasificación del tipo de consulta**

Cada pregunta vale 2 Ptos. (1 Pto., por cada parte del enunciado respondida correcta y completamente.)

##### **1. Tipo de consulta 12 (para una sola tabla)**

Objetivo. Realizar consultas básicas

Procedimiento. Se proporciona un enunciado solicitando al alumno que muestre los datos sobre una determinada tabla de la base de datos.

Enunciado (ejemplo):

- Obtener todos los datos de los estudiantes
- Busca matriculas

##### **2. Tipo de consulta 4 (para una tabla)**

Objetivo. Consultar campos de una tabla

Procedimiento. Se proporciona un enunciado solicitando al participante que muestre los datos de uno o más campos específicos de una tabla.

Enunciado (ejemplo):

- Mostrar dni nombre y paterno de todos los docentes
- Dame nombre de alumno junto con su dirección o teléfono

##### **3. Tipo de consulta 10a (para una tabla)**

Objetivo. Realizar consultas de tablas con condiciones (con elementos explícitos).

Procedimiento. Se suministra un enunciado solicitando al participante los datos (todos) de una tabla siempre y cuando cumpla determinada condición.

Enunciado (ejemplo):

- Lista los alumnos cuyo código no sea menor que 180-15 ni con código mayor a 190-15
- Encuentra estudiantes con nacimiento entre 1/1/1997 y 31/12/1997

##### **4. Tipo de consulta 10b (para una tabla)**

Objetivo. Realizar consultas de tablas con condiciones (con elementos implícitos).

Procedimiento. Se suministra un enunciado solicitando al participante los datos (todos) de una tabla siempre y cuando cumpla determinada condición.

Enunciado (ejemplo):

- Mostrar los datos de la alumna 'princesa'
- Dame los alumnos con código mayor a '180-15' menor a '190-15' y que no sea igual a 'princ' y también teléfono

### **5. Tipo de consulta 2 (para una tabla)**

Objetivo. Realizar consultas de tablas con condiciones.

Procedimiento. Se suministra un enunciado solicitando algunos datos de una tabla, siempre y cuando cumpla determinada condición.

Enunciado (ejemplo):

- Listar los nombres paternos y maternos del alumno con código 187-15
- Dame los teléfonos de los alumnos con código mayor a 180-15 pero menor a 190-15 y cuyo nombre no sea igual a PRINCESA

### **6. Tipo de consulta 0 (para una sola tabla)**

Objetivo. Realizar consultas básicas.

Procedimiento. Hacer una búsqueda mediante términos clave.

Enunciado (ejemplo):

- Estudiantes "gutierrez"
- Promedios notas estudiantes

### **7. Tipo de consulta 2 (para más de una tabla)**

Objetivo. Realizar consultas básicas.

Procedimiento. Hacer una búsqueda con campos explícitos, con condiciones y sin funciones.

Enunciado (ejemplo):

- Dame el nombre del día en que enseña el docente de nombre José
- Dame el código del alumno cuyo distrito sea Tarata

### 8. Tipo de consulta 4 (para más de una tabla)

Objetivo. Realizar consultas con campos seleccionados, con tablas, con condiciones y sin funciones.

Procedimiento. Hacer una búsqueda con campos explícitos, con condiciones y sin funciones.

Enunciado (ejemplo):

- Dame nombre de los alumnos con departamento Tacna
- Dame el nombre de los alumnos cuyo departamento sea igual a tacna

### 9. Tipo de consulta 10a y 10b (para más de una tabla)

Objetivo. Realizar consultas sin campos seleccionados, con tablas, condiciones y sin funciones.

Procedimiento. Realizar consultas sin campos, con tablas y condiciones (con columnas explícitas) y sin funciones.

Enunciado (ejemplo):

- Dame el alumno con nombre de distrito tarata
- Dame departamentos de los alumnos con distrito= Tarata

### 10. Tipo de consulta 12 (para más de una tabla)

Objetivo. Realizar consultas con campos seleccionados, con tablas, sin condiciones y sin funciones.

Procedimiento. Realizar consultas sin campos, con tablas y condiciones (con columnas explícitas) y sin funciones.

Enunciado (ejemplo):

- Dame provincias, distritos y departamentos de los alumnos
- Dame departamentos de los alumnos

**ANEXO 5. CATÁLOGO DE REQUISITOS****Requisitos funcionales de usuario**

<b>Identificador</b>		<b>RU-001</b>
Nombre	Acceder a la ILN	
Descripción	El usuario proporciona nombre y contraseña por defecto, que permite conectarse a la base de datos y así usar la ILN	
Prioridad	Alta	
Estabilidad	Alta	
Claridad	Alta	
Verificabilidad	Alta	

<b>Identificador</b>		<b>RU-002</b>
Nombre	Escribir consulta de recuperación usando lenguaje natural	
Descripción	El usuario hace sólo consultas que recuperan datos de la base de datos mediante el uso del lenguaje natural	
Prioridad	Alta	
Estabilidad	Alta	
Claridad	Alta	
Verificabilidad	Alta	

<b>Identificador</b>		<b>RU-003</b>
Nombre	Escribir consulta de recuperación usando términos clave	
Descripción	El usuario hace sólo consultas que recuperan datos de la base de datos mediante el uso de términos clave	
Prioridad	Alta	
Estabilidad	Alta	
Claridad	Alta	
Verificabilidad	Alta	

<b>Identificador</b>		<b>RU-004</b>
Nombre	Escribir consulta de recuperación usando lenguaje natural y términos clave mezclados	
Descripción	El usuario hace sólo consultas que recuperan datos de la base de datos mediante el uso mezclado de lenguaje natural y términos clave	

Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---



---

<b>Identificador</b>	<b>RU-005</b>
----------------------	---------------

---

Nombre	Visualizar la query SQL resultante
Descripción	La ILN genera y presenta al usuario una query SQL, equivalente semánticamente a la consulta que hizo el usuario
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---



---

<b>Identificador</b>	<b>RU-006</b>
----------------------	---------------

---

Nombre	Visualizar datos recuperados
Descripción	Se presentan al usuario, por medio de la interfaz gráfica, los datos recuperados de la base de datos, mediante la query SQL en la que se transformó su consulta
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---



---

<b>Identificador</b>	<b>RU-007</b>
----------------------	---------------

---

Nombre	Visualizar mensajes de advertencia
Descripción	Se presentan al usuario, mediante la interfaz gráfica, mensajes que le advierten cuando la query SQL no logra obtener resultados de la base de datos
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RU-008</b>
Nombre	Visualizar mensajes de error
Descripción	Mediante la interfaz gráfica, se presentan al usuario mensajes que le indican que no se puede traducir la consulta del usuario a una query SQL correcta
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

### Requisitos no funcionales de usuario

<b>Identificador</b>	<b>RU-009</b>
Nombre	Propósito netamente formativo
Descripción	En principio la aplicación está destinada al aprendizaje del SQL
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RU-010</b>
Nombre	Facilidad de uso
Descripción	La ILN puede ser utilizada por cualquier tipo de usuario
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RU-011</b>
Nombre	Rapidez de aprendizaje
Descripción	El usuario requiere un corto periodo de formación para utilizar la ILN
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RU-012</b>
Nombre	Ocultar procesamiento de la consulta del usuario
Descripción	El usuario no interviene en el proceso que traduce su consulta a SQL, ni en la ejecución sobre la base de datos
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

### Requisitos funcionales de software

<b>Identificador</b>	<b>RS-001</b>
Nombre	Iniciar sesión de trabajo
Descripción	La ILN acepta/rechaza la conexión a la base de datos basándose en el nombre y contraseña introducidos por el usuario al querer acceder
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RS-002</b>
Nombre	Leer consulta escrita por el usuario
Descripción	La ILN permite introducir consultas de información a la base de datos, a través de un formulario exclusivo para ese propósito con una caja de texto de consulta
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RS-003</b>
Nombre	Esperar consulta correcta por parte del usuario
Descripción	Las consultas deben cumplir ciertos requisitos: como que deben tener información suficientemente clara para poder responderse, y no deben ser ambiguas o cíclicas. Además, la información solicitada a la

base de datos debe estar antes que la condición. También, los valores literales de cadenas deben escribirse entre comillas (simples o dobles). Las fechas con un formato dd/mm/aaaa o dd/mm/aa, y no separar un mismo número con espacios

Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RS-004</b>
Nombre	Permitir sólo recuperar datos
Descripción	La ILN sólo procesa consultas que permiten recuperar datos de la base de datos, y se eluden consultas o partes de estas que no sirvan para tal efecto
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RS-005</b>
Nombre	Formatear la consulta introducida por el usuario
Descripción	Eliminar signos de puntuación, acentos y palabras superfluas. Convertir horas a formato universal, modificar condiciones con signos matemáticos y condensar literales con más de una palabra
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RS-006</b>
Nombre	Traducir consulta del usuario a una query SQL
Descripción	La ILN genera queries semánticamente equivalentes a la consulta introducida por el usuario
Prioridad	Alta
Estabilidad	Alta

Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RS-007</b>
Nombre	Combinar tablas automáticamente
Descripción	Una consulta admite involucrar a más de una tabla a la vez, mediante una forma de consulta SQL llamada combinación, que la ILN hace automáticamente
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---



---

<b>Identificador</b>	<b>RS-008</b>
Nombre	Formatear la query SQL resultante
Descripción	La query SQL generada se formatea antes de su presentación para mejorar su visualización
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---



---

<b>Identificador</b>	<b>RS-009</b>
Nombre	Mostrar la query SQL resultante
Descripción	La ILN escribe la query SQL que genera en su ventana principal para ser visualizada por parte del usuario
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---



---

<b>Identificador</b>	<b>RS-010</b>
Nombre	Abrir conexión a la base de datos
Descripción	La ILN se conecta a la base de datos con la que trabaja

Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RS-011</b>
Nombre	Recuperar información de la base de datos con la query SQL
Descripción	La ILN consulta la base de datos mediante la query SQL que generó para recuperar la información solicitada por el usuario, y así demostrar si dicha query es correcta
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RS-012</b>
Nombre	Formatear los datos recuperados
Descripción	Los datos recuperados por la consulta se formatean debidamente con un diseño predefinido que se aplica antes de ser exhibidos
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RS-013</b>
Nombre	Mostrar los datos consultados por el usuario
Descripción	Los datos recuperados se presentan al usuario en formato tabular y formateados
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

<b>Identificador</b>	<b>RS-014</b>
Nombre	Aplicar modelo E/R en la base de datos
Descripción	La base de datos debe ser relacional y debe cumplir con el modelo Entidad-Relación
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RS-015</b>
Nombre	Unicidad de nombres de tablas de la base de datos
Descripción	La descripción de una tabla debe ser distinta de la de cualquier otra
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RS-016</b>
Nombre	Unicidad de campos de una misma tabla
Descripción	La descripción de un campo perteneciente a una tabla es diferente de la de cualquier otro de la misma tabla. Pero distintas tablas pueden contener la misma descripción
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RS-017</b>
Nombre	Unicidad de identificación
Descripción	El nombre y la contraseña introducidos corresponden directamente a los que posee la base de datos para conectarse con ella
Prioridad	Alta
Estabilidad	Alta

Claridad Alta

Verificabilidad Alta

**Requisitos no funcionales de software**

<b>Identificador</b>	<b>RS-018</b>
Nombre	Módulo de traducción de la consulta del usuario no complejo
Descripción	El módulo que transforma la consulta del usuario en una query SQL debe ser relativamente pequeño, autodocumentado, y fácil de entender y modificar a posteriori
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RS-019</b>
Nombre	Adhesión a recomendaciones internacionales
Descripción	Para la construcción entera de la ILN se adoptan recomendaciones y prácticas ya probadas con anterioridad en el desarrollo de software similar o relacionado
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RS-020</b>
Nombre	Diseño descrito documentariamente
Descripción	El diseño de la ILN está enmarcado por la documentación
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

<b>Identificador</b>	<b>RS-026</b>
Nombre	Documentar cambios en la ILN

Descripción	Se documenta cualquier tipo de cambio o implementación que sea de interés para el equipo (actual o futuro) de desarrollo
Prioridad	Alta
Estabilidad	Alta
Claridad	Alta
Verificabilidad	Alta

---

Dónde:

Prioridad	: Importancia o urgencia del requisito. Posibles valores: “Por defecto”, “Alta”, “Media” o “Baja”
Estabilidad	: Grado de susceptibilidad al cambio a lo largo del proyecto. Sus valores: “Alta”, “Media” o “Baja”
Claridad	: Falta de ambigüedad del requisito (“Alta”, “Media”, “Baja”)
Verificabilidad	: Facilidad de incorporar el requisito al sistema (“Alta”, “Media” o “Baja”)

**Cuadro 41.** Matriz de trazabilidad entre requisitos de usuario y requisitos de software

RS RU	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016	017	018	019	020	21	
001	X																					
002		X	X	X	X																	
003		X	X	X	X																	
004		X	X	X	X																	
005							X	X	X													
006									X	X	X	X										
007													X									
008															X	X	X					
009																		X	X			
010																				X	X	X

**Cuadro 42.** Matriz de trazabilidad entre requisitos funcionales de software y casos de uso

RS CU	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016	017
001	X																X
002		X	X	X													
003						X	X										
004									X	X	X			X	X	X	
005								X	X			X	X				

**ANEXO 6. VALIDACIÓN DE SOFTWARE POR PARTE DE LOS USUARIOS**

Objetivo: Conocer las opiniones de los operarios de la ILN.

Indicadores: Analice con cuidado cada afirmación y responda objetivamente, marcando con una X en la columna que considere más cercana a su opinión:

PARA UD. LA ILN...	Muy de acuerdo	Un poco de acuerdo	Más o menos	Un poco en desacuerdo	Muy en desacuerdo
1. Tiene buen aspecto					
2. Visualiza bien la caja de búsqueda					
3. Recibe peticiones de información concisas					
4. Las consultas se pueden formar fácilmente					
5. Se ejecuta rápidamente					
6. Tiene buen rendimiento					
7. Crea resultados completos					
8. Visualiza bien el resultado					
9. Estandariza los resultados					
10. Los resultados son 100% fiables					
11. Devuelve resultados exactos					
12. Los resultados son comprensibles					
13. Muestra indicaciones apropiadas					
14. Se aprecia cómo se generó el resultado paso a paso					
15. Exporta archivos fácilmente					
16. Tiene alta tolerancia a los errores					
17. Es muy segura					
18. Es muy fácil de operar					
19. Está bien documentada					
20. Estoy muy satisfecho con esta aplicación					
<b>TOTAL</b>					

Fuente: Basada en Factores de Calidad de McCall (Pressman, 2010)

**Ponderación:** Muy de acuerdo vale 4pts / Un poco de acuerdo 3pts / Más o menos 2pts / Un poco en desacuerdo 1pts / Muy en desacuerdo 0pts

La ponderación por usuario se calcula con:  $Puntaje\ final = \frac{\sum_{i=1}^{20} Alternativa\ marcada_i}{10}$

El resultado final se calcula mediante el promedio de las ponderaciones de todos los usuarios

**ANEXO 7. CLASIFICACIÓN DE LOS ELEMENTOS GRAMATICALES**

**Cuadro 43.** Elementos gramaticales

<b>Categoría gramatical</b>	<b>Palabra</b>	<b>Símbolo</b>
Verbo mostrar	buscar, dar, decir, enlistar, listar, mostrar, obtener, presentar, proporcionar, seleccionar, ubicar, ver, visualizar	VM
Nombre tabla	(Nombre de las tablas en la base de datos)	NT
Nombre columna	(Nombre de los campos en la base de datos)	NC
Preposición de	de, del	DE
Preposición	con, cuando, cuya, cuyas, cuyo, cuyos, donde, que	QUE
Preposición entre	Entre	ENTRE
Conjunción	y, e, pero, además	Y
Disyunción (o exclusiva)	o, u	O
Disyunción (o inclusiva)	y/o, y/u, e/o, e/u	Y/O
Negación	ni, ningún, ninguna, ningunas, ninguno, ningunos, no	NO
Condición	que	QUE
Operador	!=, <=, <>, =, ==, >=, diferente, desigual, distinta, distinto, opuesta, opuestas, opuesto, opuestos	OPER
Operador	=, ==, igual, idéntico	OPER
Operador	>, mayor	OPER
Operador	>=, mayor igual, mayor o igual, mayor/igual	OPER
Operador	<, menor	OPER
Operador	>=, menor igual, menor o igual, menor/igual	OPER
Literal fecha	'dd/mm/aaaa', "dd/mm/aaaa"	LIT
Literal hora	'hh:mm', "hh:mm"	LIT
Literal número	(Número entero o real)	LIT
Literal texto	(Texto que va entre comillas simples o dobles)	LIT

## ANEXO 8. CÓDIGO FUENTE DE ALGUNOS MÓDULOS DE LA ILN

## Parte principal del código fuente del módulo Morfológico.php

```
1 <?php
2 require_once ('../NI2Sql/config.php');
3 require_once ('../NI2Sql/diccionario.php');
4
5 class Morfológico
6 {
7     private $_consultaUsuario;
8     private $_consultaNormalizada;
9
10    function __construct ($consultaUsuario)
11    {
12        $this->_consultaUsuario = $consultaUsuario;
13    }
14
15    public function ejecutarAnálisis ()
16    {
17        $this->_parsearConsulta ();
18        $this->_preprocesarConsulta ();
19        $this->_normalizarConsulta ();
20
21        return $this->_consultaNormalizada;
22    }
23    ...
281
282    // <- - aquí va más código
283 } #end-class
284
```

## Código fuente del módulo Sintactico.php

```
1 <?php
2 require_once ('../NI2Sql/config.php');
3 require_once ('/Automata.php');
4
5 class Sintactico
6 {
7     private $_automata;
8
9     function __construct ($consultaNormalizada)
10    {
11        $this->_automata = new Automata ($consultaNormalizada);
12    }
13
14    public function ejecutarAnálisis ()
15    {
16        return $this->_automata->recorrer ();
17    }
18 } #end-class
19
```

## Código fuente del módulo Semantico.php

```

1  <?php
2  require_once ('../NI2Sql/config.php');
3  require_once ('../ConsultaSql/ConsultaSql.php');
4
5  class Semantico
6  {
7      private $_estructuraSintactica;
8      private $_consulta;
9
10     function __construct ($estructuraSintactica)
11     {
12         $this->_estructuraSintactica = $estructuraSintactica;
13         $this->_consulta = new ConsultaSql ();
14     }
15
16     public function ejecutarAnalisis ()
17     {
18         return $this->_procedimientosSql ();
19     }
20
21     private function _procedimientosSql ()
22     {
23         foreach ($this->_estructuraSintactica as $item) {
24             $nombreFuncion = $item [0];
25             $parametro = $item [1];
26
27             $this->_consulta->$nombreFuncion ($parametro);
28         }
29
30         return $this->_consulta->obtenerConsulta ();
31     }
32
33     } # end-class
34

```

## Parte del código fuente del módulo Pragmatico.php

```

1  <?php
2  require_once ('../NI2Sql/config.php');
3  require_once ('/TodasLasColumnas.php');
4  require_once ('/ColumnaPorDefecto.php');
5  require_once ('/Guion.php');
6  require_once ('../NI2Sql/Grafo/Combinacion.php');
7  require_once ('../Model/Sql/Sql.php');
8
9  class Pragmatico
10 {
11     private $_pseudoSql;
12     private $_columnas;
13     private $_tablas;
14     private $_condiciones;

```

```

15     private $_combinaciones;
16
17     public function __construct (Sql $pseudoSql)
18     {
19         $this->_pseudoSql = $pseudoSql;
20         $this->_condiciones = $this->_pseudoSql->condiciones ();
21     }
22
23     public function ejecutarAnalisis ()
24     {
25         $this->_interpretarOperadorDeLiteralCadena ();
26         $this->_interpretarIgualdadDeCadena ();
27         $this->_interpretarDesigualdadDeCadena ();
28         $this->_interpretarComodinesConsulta ();
29         $this->_obtenerCombinaciones ();
30
31         return $this->_crearNuevoSql ();
32     }
33
...     // <- - aquí va más código
202 } # end class
203

```

### Código fuente del módulo inicio.php

```

1 <?php
2 class Inicio
3 {
4     private $_reemplazosVista;
5     private $_rutaVista;
6
7     public function __construct ()
8     {
9         $this->_reemplazosVista = array (
10             'titulo' => 'Inicio del sistema',
11             'tarea' => 'Inicio',
12             'tarea-contenido' => 'Haga clic en los menús superiores '
13                 . 'para comenzar, '
14                 . 'o en "Buscar..." para hallar información.',
15             'resultados' => 'Bienvenido',
16             'sql' => "",
17             'tabla-resultados' => "",
18             'detalles-contenido' => ""
19         );
20         $this->_rutaVista = '../templates/inicio.html';
21     }
22
23     private function _obtenerTemplate ()
24     {
25         return file_get_contents ($this->_rutaVista);
26     }

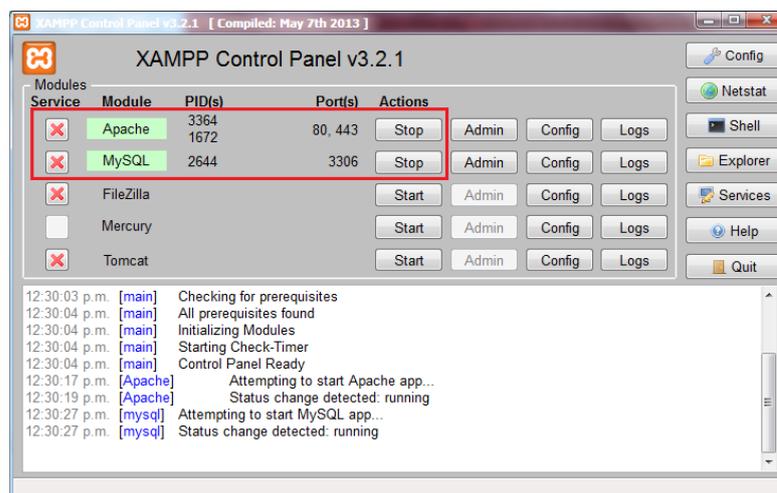
```

```
27     public function ejecutar ()
28     {
29         $html = $this->_obtenerTemplate ();
30
31         foreach ($this->_reemplazosVista as $buscado => $reemplazo) {
32             $html = str_replace ('{ ' . $buscado . ' }', $reemplazo, $html);
33         }
34
35         echo $html;
36     }
37
38     } # end-class
39
40     $inicio = new Inicio ();
41     $inicio->ejecutar ();
42
```

**ANEXO 9. BREVE MANUAL DEL USUARIO DE LA ILN**

**Comenzando con la aplicación**

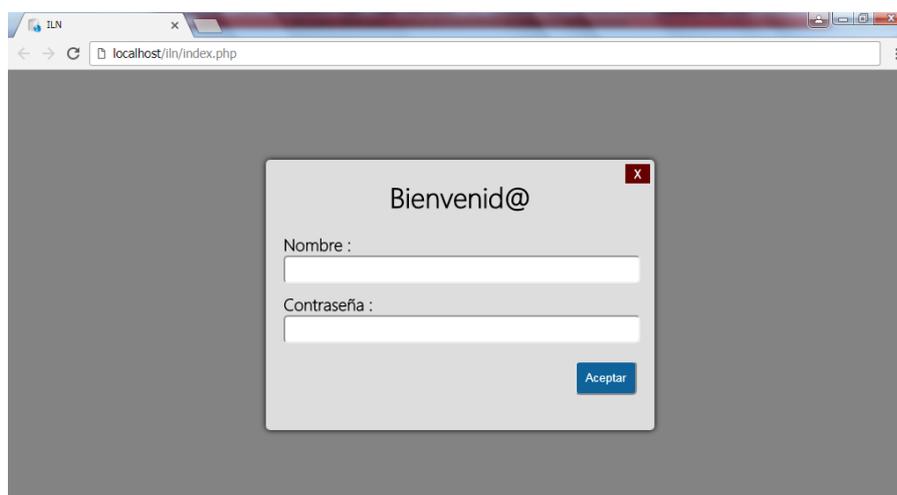
Previamente se debe activar el XAMPP, aparece la ventana del Panel de Control. En él se presiona en los botones *Start* de la columna *Actions* correspondientes a los módulos Apache y MySQL. Una vez activados las etiquetas de las columnas PID(s) y Port(s) cambian, igual que los rótulos de los antedichos botones poniéndose a *Stop*.



Luego, se abre un navegador web para comenzar a trabajar con la ILN. En la caja de dirección del navegador debe introducirse la siguiente dirección: <http://localhost/iln/index.php>, y presionar Enter.



Aparece la ventana de autenticación. Siempre que se inicie la aplicación aparecerá dicha ventana. Se llenan los campos de texto con los datos respectivos, y luego se hace clic en Aceptar o se presiona Enter otra vez.



Si los datos no son correctos la ventana cambia de aspecto e informa del error.

X

## Bienvenid@

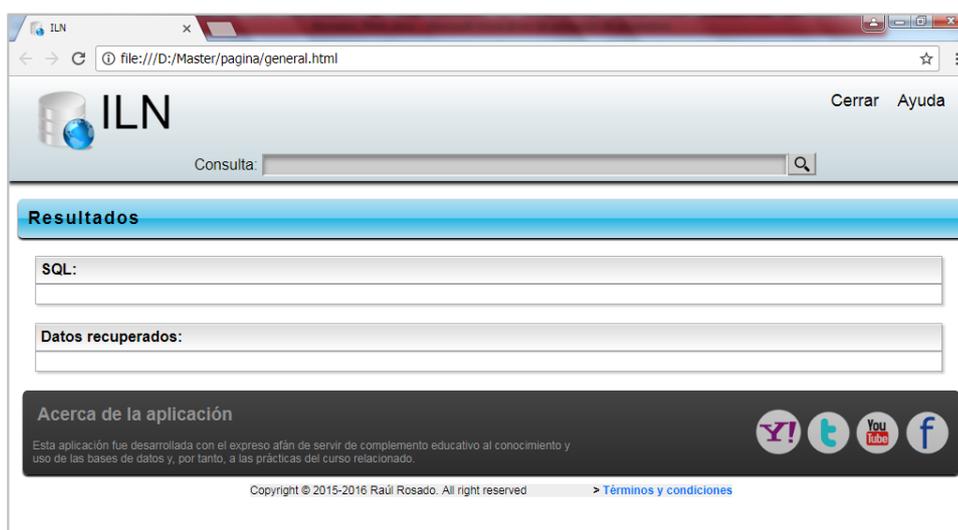
Nombre :

Contraseña :

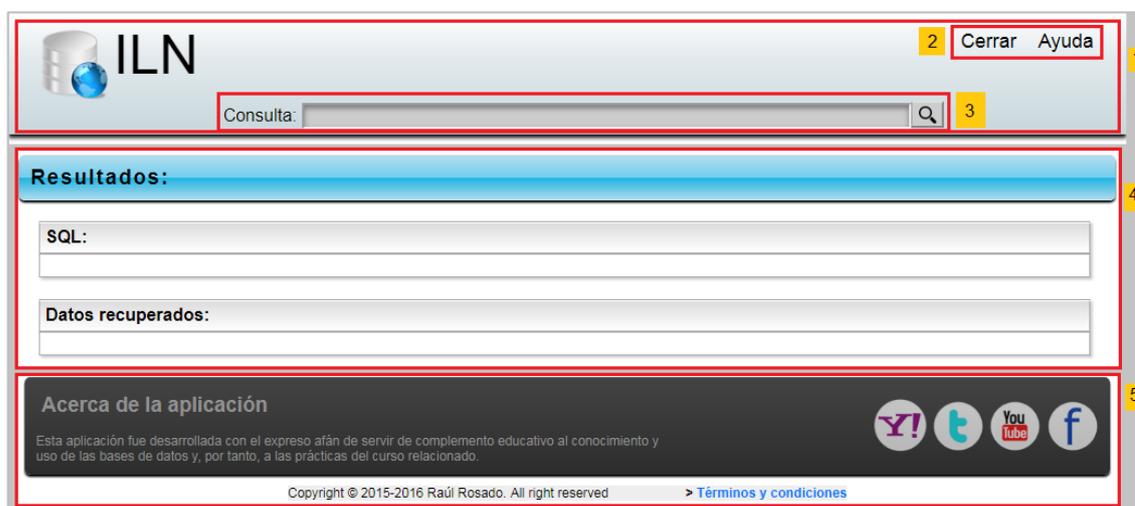
Aceptar

Error: el nombre o la contraseña ingresados no son correctos. Por favor vuelva a ingresar dichos datos.

En la parte superior aparece un botón marcado con una X que sirve para cancelar el proceso de autenticación y abandonar la aplicación. Este botón se puede presionar en cualquier momento durante dicho proceso. Si los datos ingresados son correctos aparecerá la ventana principal de la ILN (mostrada a continuación).



### Conociendo la ILN



En la parte superior puede apreciarse la sección del título principal señalada con el número 1, la cual contiene la caja de texto (número 3) donde el usuario podrá ingresar la consulta en lenguaje natural o términos clave, o ambos para recuperar información. También contiene al menú con dos opciones “Cerrar” y “Ayuda” (número 2). Más abajo está la sección “Resultados:” (número 4), donde se mostrarán los resultados de la consulta. Por último está la sección “Acerca de la aplicación” (número 5) con cuatro iconos de contacto, donde se podrá obtener más información acerca de la ILN, términos y condiciones de uso, y su desarrollador.

### Recuperando información

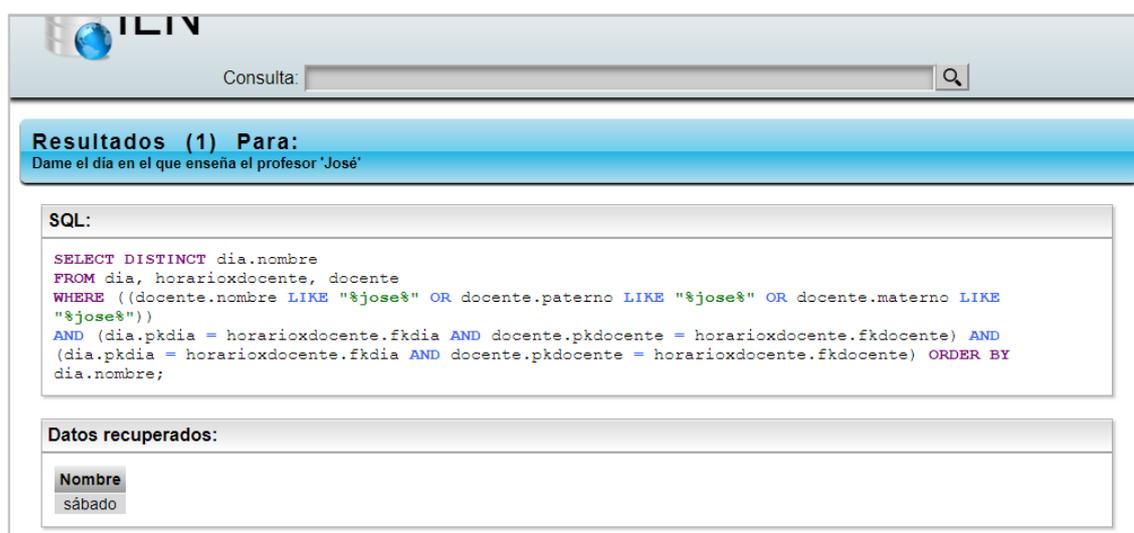
Para realizar esta operación tan sólo escriba su consulta en lenguaje natural en la caja de búsqueda señalada con el número 1 en la imagen siguiente.

Nota.- recuerde que al escribir texto debe hacerlo lo mejor posible, es decir, sin errores de sintaxis ni gramaticales. Recuerde también que el software ILN es tolerante con ciertas faltas ortográficas como los acentos.



Luego presione Enter o haga clic sobre el icono de búsqueda (una pequeña lupa) al costado de la caja de búsqueda (señalada con el número 2 en la imagen precedente).

La ILN responde mostrando los resultados de la consulta hecha.



Nota.- El título **Resultados:** cambia indicando que hay un número ‘n’ de respuestas que satisfacen la consulta recientemente ingresada y procesada; además, se transcribe la consulta del usuario debajo.

Nota.- El recuadro etiquetado con **SQL**: cambia mostrando la sentencia que consulta la base de datos (en lenguaje SQL) generada automáticamente por la ILN, con sus partes debidamente formateadas para una mejor visualización.

Nota.- El recuadro etiquetado con **Datos recuperados**: también cambia para mostrar los datos que se recuperaron por la ILN usando automáticamente la sentencia SQL generada, de modo que se puede verificar al mismo tiempo si es correcta o no.

Nota.- Los resultados siempre se muestran en forma de tabla.

En caso de que aparezcan varios resultados estos se presentarán en forma de tabla:

**Resultados (10) Para:**  
 Muestrame a los profesores

**SQL:**  

```
SELECT DISTINCT docente.nombre, docente.paterno, docente.materno, docente.doc_identidad, docente.sexo,
docente.domicilio, docente.telefono, docente.contacto
FROM docente
ORDER BY docente.nombre, docente.paterno, docente.materno, docente.doc_identidad, docente.sexo,
docente.domicilio, docente.telefono, docente.contacto
```

**Datos recuperados:**

Nombre	Paterno	Materno	Doc_identidad	Sexo	Domicilio	Telefono	Contacto
ernesto	inofuentes	ramirez	30162799	h	jr chambilla 241	975221891	ernest_in@hotmail.com
felipe	quispe	mamani	01887191	h	jr lima 647	911785611	
gladys	ramos	duenas	30127695	m	jr benigno ballon 304	975449812	grd_istpj@gmail.com
jorge	ramos	surco	01294877	h	jr asuncion 304		jramoss@hotmail.com
jose	arteta	castillo	40219812	h	jr san bartolome 125	993018512	artetajose@gmail.com
justino	vilca	laura	40217891	h	jr arequipa 310	978552163	facebookjustino_vila
mario	huayta	molina	41227005	h	av alfonos ugarte 331	988782391	meli_hg@gmail.com
rosario	ponce	yupanqui	02398119	m	jr ilave 674	975159812	charo.ponce@gmail.com

### Consultas sin respuestas

Puede ser que en ocasiones se ingrese alguna consulta que no dé ningún resultado. En ese caso la ILN muestra un mensaje apropiado. Esto significa que la base de datos no responde a esa consulta, ya sea porque no existe ese elemento en la base de datos, ya porque la consulta está mal planteada, etc. Los elementos **SQL**: y **Datos recuperados**: posiblemente no muestre ningún tipo de información o lo haga de manera inconclusa. El título **Resultados** mostrará el número cero (0), indicando que no hay nada que mostrar.

**Resultados (0) Para:**  
 estudiante con apellido "Machicao"

Ningún registro satisface la consulta

**SQL:**  

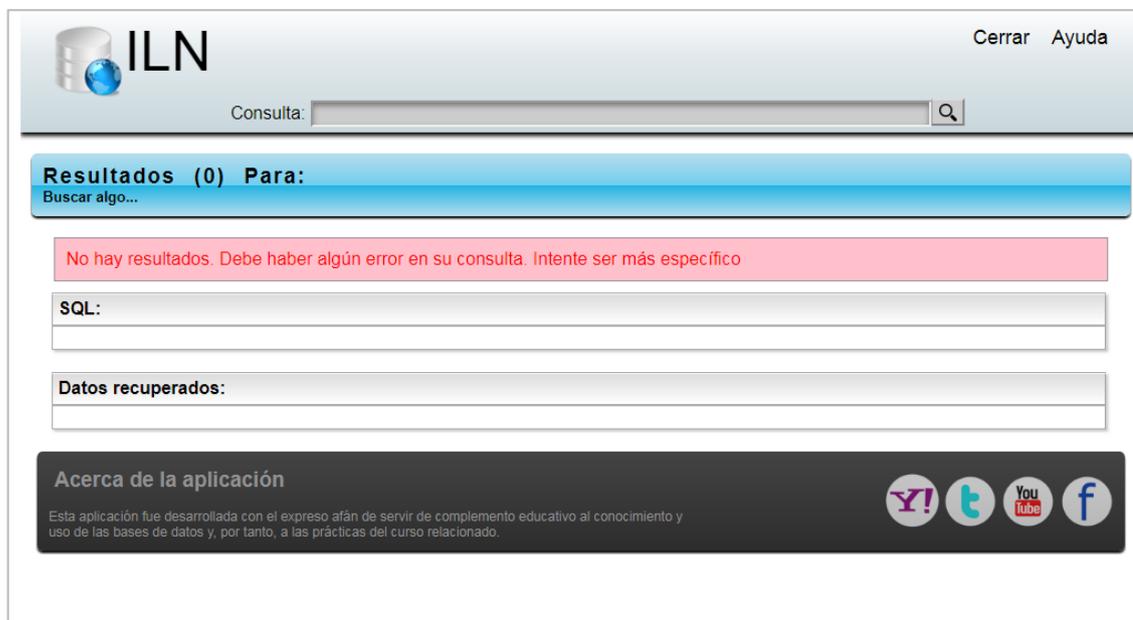
```
SELECT DISTINCT alumno.codigo, alumno.nombre, alumno.paterno, alumno.materno, alumno.doc_identidad,
alumno.sexo, alumno.estado_civil, alumno.nacimiento, alumno.domicilio, alumno.telefono,
alumno.contacto
FROM alumno
WHERE alumno.nombre LIKE "%machicao%" ORDER BY alumno.codigo, alumno.nombre, alumno.paterno,
alumno.materno, alumno.doc_identidad, alumno.sexo, alumno.estado_civil, alumno.nacimiento,
alumno.domicilio, alumno.telefono, alumno.contacto
```

**Datos recuperados:**

--

### Consultas con errores

Puede ser que en ocasiones ingrese alguna consulta incorrecta. En tal caso la interfaz muestra claramente ese error. Los elementos **SQL:** y **Datos recuperados:** posiblemente no muestren ningún tipo de información, o lo haga de manera inconclusa.



The screenshot shows the ILN application interface. At the top, there is a header with the ILN logo and a search bar labeled "Consulta:". To the right of the search bar are buttons for "Cerrar" and "Ayuda". Below the search bar, there is a section titled "Resultados (0) Para:" with a sub-label "Buscar algo...". A red error message box states: "No hay resultados. Debe haber algún error en su consulta. Intente ser más específico". Below this, there are two empty input fields labeled "SQL:" and "Datos recuperados:". At the bottom, there is a dark grey footer section titled "Acerca de la aplicación" with a small text description and social media icons for YouTube, Twitter, and Facebook.

### Buscando ayuda

Se puede obtener ayuda sobre la ILN en dos partes del programa: en la opción Ayuda. Al hacer clic sobre ese botón aparece una página web con el contenido del manual. El segundo lugar es haciendo clic en alguno de los iconos de la sección "Acerca de la aplicación", donde si se tiene conexión a Internet, se podrán escribir y leer ayudas sugerencias preguntas o respuestas, o ver tutoriales.

### Salir de la ILN

Para salir de la ILN basta con hacer clic en la opción Cerrar del menú principal. Este cierra la conexión a la base de datos y cambia la ventana principal por la ventana de Autenticación para que otro usuario o el mismo puedan volver a empezar el trabajo con la ILN si así se deseara.

**ANEXO 10. CLASIFICACIÓN DEL TIPO DE CONSULTA**

**Cuadro 44.** Tipos de consulta

Elementos Tipo	Campos	Tablas	Condiciones	Funciones	Calificación	Observación
Tipo 1	Presente	Presente	Presente	Presente	Simple	
Tipo 2	Presente	Presente	Presente		Simple	Implementado
Tipo 3	Presente	Presente		Presente	Simple	
Tipo 4	Presente	Presente			Simple	Implementado
Tipo 5	Presente		Presente	Presente	Compleja	
Tipo 6	Presente		Presente		Compleja	
Tipo 7	Presente			Presente	Compleja	
Tipo 8	Presente				Compleja	
Tipo 9		Presente	Presente	Presente	Simple	
Tipo 10		Presente	Presente		Simple	Implementado
Tipo 11		Presente		Presente	Simple	
Tipo 12		Presente			Simple	Implementado
Tipo 13			Presente	Presente	Compleja	
Tipo 14			Presente		Compleja	
Tipo 15				Presente	Compleja	
Tipo 16					Compleja	

Fuente: Basada en Pazos Rangel (2010)

Nota 1. - Existe un tipo de consulta 0, que se refiere al uso de palabras clave

Nota 2.- Las funciones son "HAVING", "GROUP BY", "ORDER BY", "MAX", "MIN", "SUM", "AVG", "COUNT", etc.

Nota 3.- Dentro del tipo 10 existen todavía dos subtipos: cuando los campos de la condición son explícitos, y cuando no lo son.

**ANEXO 11. MATRIZ DE CONSISTENCIA**

Aplicación de una Interfaz de Lenguaje Natural en el uso de la base de datos relacional del Sistema de Registro Académico Instituto de Educación Superior Tecnológico Público Juli, 2014

PROBLEMA	HIPÓTESIS	OBJETIVO	VARIABLE	DIMENSIÓN	INDICADOR	MÉTODO
¿En qué medida mejoran en el uso de las bases de datos relacionales los usuarios del Sistema de Registro Académico del Instituto de Educación Superior Tecnológico Público Juli con la aplicación de una interfaz de lenguaje natural?	La mayoría de los usuarios del Sistema de Registro Académico del Instituto de Educación Superior Tecnológico Público Juli mejoran significativamente en el uso de las bases de datos relacionales con la aplicación de una interfaz de lenguaje natural	Determinar la medida en que mejoran en el uso de bases de datos relacionales los usuarios del Sistema de Registro Académico del Instituto de Educación Superior Tecnológico Público Juli con la aplicación de una interfaz de lenguaje natural	Interfaz de lenguaje natural	Desarrollo de software orientado a objetos	- Análisis - Diseño - Implementación - Prueba	Técnica: - Métricas de software Instrumento: - McCall - Puntos de función - Diseño - Halstead Técnica: - Examen práctico Instrumento: - Prueba de performance
¿Cuál es el nivel que alcanza el análisis orientado a objetos respecto de una interfaz de lenguaje natural para el uso de una base de datos relacional?	El análisis orientado a objetos respecto de una interfaz de lenguaje natural para el uso de una base de datos relacional alcanza un nivel aceptable	Determinar el nivel que alcanza el análisis orientado a objetos respecto de una interfaz de lenguaje natural para el uso de una base de datos relacional	Uso de una base de datos relacional	Manejo procedimental	- Manipulación	
¿Cuál es el nivel que logra el diseño orientado a objetos respecto de una interfaz de lenguaje natural para usarla sobre una base de datos relacional?	El diseño orientado a objetos respecto de una interfaz de lenguaje natural para usarla sobre una base de datos relacional logra un nivel aceptable	Evaluar el nivel logrado por el diseño orientado a objetos respecto de una interfaz de lenguaje natural para usarla sobre una base de datos relacional				
¿En qué medida se implementa mediante programación orientada a objetos y se prueba una interfaz de lenguaje natural sobre una base de datos relacional?	La implementación mediante programación orientada a objetos y prueba de una interfaz de lenguaje natural sobre una base de datos relacional corresponden a un nivel aceptable	Señalar la medida en que se implementa mediante programación orientada a objetos y se prueba una interfaz de lenguaje natural sobre una base de datos relacional				
¿Qué nivel de manejo procedimental muestran los estudiantes del Instituto de Educación Superior Tecnológico Público Juli respecto del uso de una base de datos relacional antes y después de experimentar con una interfaz de lenguaje natural?	El nivel de manejo procedimental que muestran los estudiantes del Instituto de Educación Superior Tecnológico Público Juli respecto del uso de una base de datos relacionales antes del experimento es del nivel Deficiente; mientras que después del mismo es del nivel Bueno	Evaluar el nivel de manejo procedimental que muestran los estudiantes del Instituto de Educación Superior Tecnológico Público Juli respecto del uso de una base de datos relacional antes y después de experimentar con una interfaz de lenguaje natural				