UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERIA, ARQUITECTURA Y URBANISMO

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

ATENCIÓN DE CONSULTAS DEL USUARIO USANDO EL PROCESAMIENTO DEL LENGUAJE NATURAL EN EL ÁMBITO DE SOPORTE TÉCNICO.

PARA OPTAR TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor:
Guillermo Eduardo Lapoint Ruiz

Asesor:
Ing. José Fortunato Zuloaga Cachay

Línea de Investigación:
Inteligencia Artificial

Pimentel – Perú
2018

ATENCIÓN DE CONSULTAS DEL USUARIO USANDO EL PROCESAMIENTO DEL LENGUAJE NATURAL EN EL ÁMBITO DE SOPORTE TÉCNICO.

Aprobación de la Tesis

Ing. Valdivia Salazar Carlos Alberto
Presidente del jurado de tesis

Mg. Tuesta Monteza Víctor
Secretario del jurado de tesis

Mg. Mejia Cabrera Ivan
Vocal del jurado de tesis

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

DEDICATORIA

A mis hijos, que son el motivo por el que sigo adelante, a mi esposa por la paciencia y dedicación que tiene conmigo y con nuestros hijos. Al apoyo de la familia que, aunque lejos estén, siempre los llevo presente.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

AGRADECIMIENTO

Un agradecimiento a todas las personas que de alguna u otra forma colaboraron conmigo para el cumplimiento de este hito profesional aletargado por circunstancias de la vida.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

INDICE

Resumen *13*

Abstract *15*

CAPÍTULO I. PROBLEMA DE INVESTIGACIÓN *17*

1.1. Situación Problemática 17

1.2. Formulación del Problema 21

1.3. Delimitación de la Investigación 22

1.4. Justificación e Importancia de la Investigación. 22

1.4.1. Justificación científica. 23

1.4.2. Justificación social. 23

1.4.3. Justificación tecnológica. 23

1.5. Limitaciones de la investigación 24

1.6. Objetivos 24

CAPÍTULO II. MARCO TEÓRICO *26*

2.1. Antecedentes de la Investigación. 26

Funciones de similitud sobre cadenas de texto: Una comparación basada en la naturaleza de los datos. 26

Implementación de Técnicas de String Matching y Selección semántica aproximada en un motor de normalización terminológica. 28

Técnicas de procesamiento del lenguaje natural en la recuperación de información. 29

Creación automática de sistemas de búsqueda de respuestas en dominios restringidos. 30

Aplicaciones del procesamiento del lenguaje natural. 31

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Búsqueda de documentos basada en el uso de índices ontológicos creados por MapReduce 32
MERA: Musical Entities Reconciliation Architecture. 33
2.2. Estado del Arte 34
2.2.1. Sistemas de Pregunta-Respuesta 35
Historia de los sistemas de los sistemas de pregunta-respuesta 36
Aplicaciones de los Sistemas de Pregunta-Respuesta 39
2.2.2. Sistemas de Búsqueda de Respuesta de Dominio Restringido (SBR-DR) 40
2.2.3. String Matching en PLN 41
2.3. Bases Teórico – Científicas 42
2.3.1. Mesa de ayuda 42
1. ¿Qué es una Mesa de ayuda? 42
2. Mesa de ayuda en las empresas 43
2.3.2. Inteligencia Artificial 44
1. Definición de Inteligencia artificial 44
2. Resumen histórico de la IA. 45
3. Tecnologías relacionadas a la Inteligencia Artificial 51
4. Aplicaciones de la Inteligencia Artificial 52
5. Procesamiento del Leguaje Natural 52
I. Revisión histórica 53
II. Concepto de procesamiento del lenguaje natural 55
III. Cuatro niveles de análisis 57
Análisis Morfológico 57
Análisis Sintáctico 58
Análisis Semántico 59
Análisis Pragmático 62
Analizadores Sintácticos 62
Procedimientos de análisis 63

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Analizadores descendentes 63
Analizadores Ascendentes 64
Algunas aplicaciones del PLN 65
IV. Técnicas de Análisis del lenguaje 66
o Técnicas Lingüísticas formales. 67
o Técnicas Probabilísticas: 68
o Pre procesamiento de la consulta: 69
a. Bag of words: 69
b. Stemming: 69
c. Lematización: 69
d. Eliminación de palabras de paso o stopwords: 70
e. Análisis sintáctico superficial. 70
f. Análisis sintáctico de dependencias. 70
g. Desambiguación del sentido de las palabras. 71
h. Obtención de representaciones semánticas. 71
i. Sistema de clasificación de la pregunta. 71
j. Obtención del foco de la pregunta. 71
k. Extracción de los términos clave de la pregunta. 72
o Recuperación de la información: 72
a. Recuperación de información sobre datos estructurados: 73
b. Recuperación de información sobre datos no estructurados o semi estructurados: 73
o Extracción de la respuesta: 74
a. Análisis de los pasajes relevantes: 74
b. Extracción de la respuesta: 74
c. Validación de la respuesta: 75

CAPÍTULO III. Marco Metodológico 76
3.1. Tipo de estudio 76
3.1.1. Tipo de Investigación 76

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.1.2. Diseño de la investigación 76
3.2. Población y Muestra 76
3.3. Hipótesis 78
3.4. Operacionalización 78
3.5. Métodos, técnicas e instrumentos de recolección de datos 78
3.6. Plan de análisis estadístico de datos 78
3.7. Análisis Estadístico e Interpretación de los datos 79
3.8. Criterios éticos 79
3.9. Criterios de rigor científico 79

CAPÍTULO IV. ANALISIS E INTERPRETACIÓN DE LOS RESULTADOS 81
4.1. Resultados. 81
4.2. Discusión de resultados. 91
4.2.1. En relación a un antecedente.- 91
4.2.2. En relación a las bases teóricas.- 92

CAPÍTULO V. PROPUESTA DE INVESTIGACIÓN 93
5.1. Selección de la técnica de procesamiento de lenguaje natural. 93
5.2. Diseñar el método de procesamiento de consultas de lenguaje natural. 96
5.2.1. Arquitectura de la aplicación. 96
5.2.2. Tecnologías empleadas. 97
- Netbeans IDE 8.0.2: 97
- Lenguajes de Desarrollo: 97
- Apache Tomcat: 98
- PostgreSQL: 98

5.2.3. Metodología de desarrollo 98
2. Fase de Elaboración: 102
A. Requerimientos 102
i. Requerimientos Funcionales 102
ii. Requerimientos No funcionales 103

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

iii. Modelo de casos de uso de Requerimiento (MCUR) 104
B. Diagrama de colaboración 110
C. Diagrama de Secuencia 111
3. Fase de Construcción: 115
A. Diagrama de Clases 115
B. Diagrama de Base de Datos 115
C. Diagrama de Componentes 116
Componentes del Sistema 116
Componentes del Algoritmo PLN. 117
D. Diagrama de Despliegue 118
E. Interfaces del Sistema 118
Inicio de Sesión del Sistema 118
Página principal 119
Interface Mantenimiento Empresa 120
Interface Mantenimiento Personal 121
Interface Mantenimiento Usuario 123
Interface Mantenimiento Problema, Causa y Soluciones. 125
Consultar solución por PLN 126
5.3. Detalle técnico del método de procesamiento de consultas. 127
5.3.1. Algoritmo LCS (Longest Common subsequence). 127
5.3.2. Algoritmo Levenshtein. 130
5.3.3. Integración de los algoritmos LCS y Levenshtein. 133
5.4. Implementar en lenguaje de programación el método de procesamiento de consultas. 135
5.4.1. Analizador de Consulta. 135
5.4.2. Selección de Sub conjunto de datos 143
5.4.3. Lógica Levenshtein 147
CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES 152
6.1. Conclusiones. 152

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

6.2. Recomendaciones. 153
Referencias 155

INDICE DE TABLAS

Cuadro 1 : Resumen histórico de la Inteligencia Artificial, explicando los datos mas relevantes de las investigaciones. 45
Cuadro 2 : Operacionalización de las variables. Presenta las variables a medir. 78
Cuadro 3 : Resultado de Tiempos transcurrido por consulta. 81
Cuadro 4 : Resultado de Tiempos Promedio de consultas. 84
Cuadro 6 : Resultado de precisión de consultas. 87
Cuadro 7 : Resultado de métrica de cadenas. 94
Cuadro 8 : Descripción especificación del caso de uso del negocio. 100
Cuadro 9 : Descripción Requerimientos funcionales. 102
Cuadro 10 : Descripción Requerimientos No funcionales. 104
Cuadro 11 : Descripción Caso de Uso: Registrar Empresa. 105
Cuadro 12 : Descripción Caso de Uso: Registrar Usuario. 106
Cuadro 13 : Descripción Caso de Uso: Registrar Persona. 108
Cuadro 14 : Descripción Caso de Uso: Registrar Problema, Causa y Solución. 109
Cuadro 14 : Descripción proceso LCS. 129
Cuadro 15 : Descripción Código Analizador de Consulta. 136
Cuadro 16 : Descripción Código Sub Conjunto de Datos. 143
Cuadro 17 : Descripción Código Lógica de Levenshtein. 147

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

INDICE DE ILUSTRACIONES

Figura 1: Ejemplo árboles de representación del análisis. 59
Figura 2 : Gráfico de Tiempos transcurrido por consulta 85
Figura 3 : Pantalla que muestra una búsqueda y su resultado. 86
Figura 4 : Pantalla que muestra el proceso de selección de la búsqueda y su resultado. 87
Figura 5 : Gráfico de porcentajes de precisión de consultas realizadas. 90
Figura 6 : Arquitectura de la solución Fuente: Diseño propio. 96
Figura 7 : Caso de uso del negocio 99
Figura 8 : Diagrama Gestión de Soporte 101
Figura 9 : Diagrama Dominio del Problema 102
Figura 10 : Diagrama Casos de uso de requerimiento. 105
Figura 11 : Diagrama de Colaboración: Registrar Empresa. 110
Figura 12 : Diagrama de Colaboración: Registrar Usuario. 110
Figura 13 : Diagrama de Colaboración: Registrar Persona. 111
Figura 14 : Diagrama de Colaboración: Registrar Problema, Causa y Solución. 111
Figura 15 : Diagrama de Colaboración: Registrar Empresa. 112
Figura 16 : Diagrama de Colaboración: Registrar Usuario. 113
Figura 17 : Diagrama de Colaboración: Registrar Persona. 114
Figura 18 : Diagrama de Colaboración: Registrar Problema, Causa y Solución. 114
Figura 19 : Diagrama de Clases. 115
Figura 20 : Diagrama de Base de datos. 116
Figura 21 : Diagrama de Componentes del Sistema. 117
Figura 22 : Diagrama del Algoritmo PLN. 117
Figura 23 : Pantalla de inicio de sesión. 118
Figura 24 : Pantalla de Pantalla principal. 119
Figura 25 : Pantalla de Mantenimiento Empresa. 120

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 26 : Pantalla de Mantenimiento Personal - Vista. 121
Figura 27 : Pantalla de Mantenimiento Personal - Agregar. 122
Figura 28 : Pantalla de Mantenimiento Usuario - Vista. 123
Figura 29 : Pantalla de Mantenimiento Usuario - Agregar. 124
Figura 30 : Pantalla de Mantenimiento Problema, Causa y Soluciones. 125
Figura 31 : Pantalla de Consulta de solución por PLN. 126
Figura 32 : Ejemplo de Pseudocódigo – Algoritmo Levenshtein. 131
Figura 33: Ejemplo de Pseudocódigo – Algoritmo Levenshtein – Sec. 1 132
Figura 34: Ejemplo de Pseudocódigo – Algoritmo Levenshtein – Sec. 2 132
Figura 35 : Gráfico de Integración LCS - Levenshtein. 133

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Resumen

En este proyecto de investigación se presenta un sistema de búsqueda de respuesta que busca procesar adecuadamente las consultas del usuario en lenguaje natural basada en texto para mejorar el tipo de respuesta esperada en el ámbito de soporte técnico. El problema que afronta esta investigación es buscar la mejor técnica que permita obtener una comprensión a nivel textual de este tipo de atenciones y brindar respuestas en tiempos reducidos y con un mayor nivel de precisión, es por ello que el objetivo se puede definir con el siguiente texto "procesar adecuadamente las consultas del usuario en lenguaje natural basada en texto para mejorar el tipo de respuesta esperada en el ámbito de soporte técnico". Para cumplir con este objetivo se estudiaron diversas técnicas de procesamiento de lenguaje natural, que pasaron desde las técnicas ontológicas hasta la de búsqueda en corpus, de las cuales se tuvo que seleccionar la técnica que mejor se adecuó a la investigación tomando la de Levenshtein para aplicar a la misma a la cual se le complementó con otro algoritmo (LCS) que mejoró el motor de búsqueda y que hizo de ésta una herramienta evolucionada por sus características. Una vez establecida la base teórica se describe el diseño de la herramienta considerando que se realizó usando las metodologías RUP y UML orientada a objetos, así como también se hace una descripción técnica detallada de los algoritmos usados mostrando el código relevante o Core de la herramienta del motor de búsqueda y se hace una descripción de la herramienta implementada como una aplicación web.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Los resultados experimentales son alentadores ya que se logró hacer uso de esta técnica con tiempos de respuesta de un promedio de 113.93 milisegundos dependiendo de la complejidad de la consulta y acercamiento a la pregunta almacenada teniendo nuestro motor de búsqueda una precisión del 93.33%. Por lo tanto, se puede concluir mencionando que esta investigación brinda aportes significativos en la aplicación este tipo de herramientas de este campo de estudio que aún no toma la relevancia que amerita.

Palabras Clave

Procesamiento de lenguaje natural, Atención de consulta de usuarios.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Abstract

In this research project presents a Response search system that seeks to process properly user queries in natural language based text to improve the type of response expected in the area of technical support. The problem facing this research is to seek the best technique that allows to obtain an understanding of a textual level of this type of attentions and provide answers in reduced times and with a higher level of presition, is why the objective can be defined with the next text "to properly process user queries in the natural language in the text to improve the type of response expected in the technical support area." To meet this objective, several techniques of natural language processing were studied, what happened from the ontological techniques to the search in corpus, from which the technique was selected that better adapted an investigation of the Levenshtein take to apply, to which it was complemented by another algorithm (LCS) that improved the search engine and made it a tool evolved by its characteristics. Once the theoretical basis has been established to describe the design of the tool that was done using the RUP and object-oriented UML methodologies, as well as a detailed technical description of the algorithms used is made, showing the relevant code or body of the search engine tool and a description of the Tool implemented as a web application.

The experimental results are encouraging since it was posible to make use of this technique with response times of an average of 113.93 milliseconds depending on the complexity of the query and the approach to the stored query having our search engine an accuracy of 93.33%. Therefore, it can conclude by mentioning that this

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

research offers significant contributions in the application of this type of tools of this field of study that still does not take the relevance it deserves.

Key Words

Natural Language processing, Attention to user queries.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO I. PROBLEMA DE INVESTIGACIÓN

1.1. Situación Problemática

Una Mesa de ayuda es un área de una empresa que responde a las preguntas técnicas de los usuarios. La mayoría de las grandes empresas tienen Mesas de ayuda para responder a las preguntas de los usuarios. Las respuestas a estas preguntas pueden ser entregadas por teléfono, correo, chat o boletines.

Los problemas técnicos de hardware o software son muy comunes hoy en día en las organizaciones, esto conlleva a solicitar ayuda al equipo de “Help Desk”, que en muchos de los casos atienden de acuerdo al orden de llegada o al criterio de priorización que tengan de acuerdo las políticas establecidas, lo cual trae como consecuencia pérdida de tiempo (Wooten & Wooten, 2001).

Muchas veces los usuarios buscan soluciones más ágiles que le permitan continuar con sus responsabilidades, dentro de estas, es buscar ayuda en los buscadores de internet (Google, Yahoo, Bing, etc.) o técnicamente llamados sistemas de recuperación de información (RI); que si bien es cierto son de ayuda, brinda grandes cantidades de información que genera también pérdida de tiempo, al tener que “bucear” dentro toda la lista de páginas brindadas para poder encontrar lo que interesa resolver (Díaz, 2009).

A pesar de los esfuerzos de los investigadores que estudian los sistemas de recuperación de información (RI), estos sistemas pierden notoriedad cuando se trata de resolver casos de temas de dominio restringido (Vila, Mazón, &

Ferrández, Creación automática de sistemas de búsqueda de respuestas en dominios restringidos., 2012).

La disciplina denominada procesamiento del lenguaje natural (PLN) resulta especialmente útil en este contexto puesto que se encarga de estudiar los problemas de la generación y la comprensión automática del lenguaje natural. El PLN también se centra en diseñar sistemas y mecanismos eficaces que permitan la comunicación entre personas y máquinas.

Para ello es necesaria la implementación de sistemas de recuperación de información capaz de procesar el lenguaje natural y de “comprender” tanto las consultas que plantea el usuario como la información almacena en su base de datos. Entre los muchos tipos de sistemas que responden a esta filosofía se pueden citar los de búsqueda de respuestas (BR).

A pesar de su corta trayectoria, los sistemas BR constituyen una interesante opción de cara a la recuperación de información y a la satisfacción de las necesidades de los usuarios (Vila, Mazón, & Ferrández, Creación automática de sistemas de búsqueda de respuestas en dominios restringidos., 2012), por tanto son sistemas especialmente útiles en situaciones donde se necesita conocer un extracto específico de información relevante.

La mayoría de este tipo de sistemas en enfocan básicamente en el idioma inglés dejando relegado las investigaciones en el idioma español y demás idiomas (Olvera-Lobo & Robinson-García, 2009). Siendo lo mencionado en este párrafo una de los factores que influyen para la realización del presente estudio.

Hoy en día se cuenta en el mercado con herramientas BR comerciales, pero todas son de ámbito general que no aportan eficiencia al obtener respuestas; para poder obtener respuestas esperadas es necesario enfocarse en un dominio restringido.

De acuerdo a las estadísticas obtenidas de una serie de experimentos para comparar la precisión de un sistema de BR de dominio abierto desarrollado en la Universidad de Alicante, llamado AliQAn y su adaptación al dominio agrícola, se pudieron obtener los resultados que indicaron que la precisión del sistema de BR AliQAn fue de un 28.8% en dominio abierto comparado con los resultados obtenidos al adaptarlo a un dominio restringido que fueron de un 58.3% de precisión (Vila, Mazón, & Ferrández, Creación automática de sistemas de búsqueda de respuestas en dominios restringidos., 2012).

Los sistemas de BR de dominio restringido (SBR-DR) es un sistema diseñado para adaptarse de forma óptima a un área concreta. Dentro de estos sistemas pueden tratarse diferentes tipos de preguntas, siendo el ámbito de aplicabilidad el que determine qué tipo de preguntas resulten más interesantes (Vila, Búsqueda de respuestas en dominios restringidos: aplicación sobre el dominio agrícola., 2010).

En la actualidad los SBR-DR abarcan varios temas entre ellos se han encontrado sistemas para los ámbitos: turísticos, médico, de pronóstico del tiempo, académico entre otros (Vila, Búsqueda de respuestas en dominios restringidos: aplicación sobre el dominio agrícola., 2010), de los cuales se

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

tienen: EXTRANS, que ayuda a dar respuesta a preguntas arbitrarias sobre archivos de documentación UNIX; WEBCOOP, que da respuesta a varios aspectos de dominios turísticos parcialmente restringidos; SBR-DR para pronóstico del tiempo, que es un Robot doméstico que contesta preguntas sobre el pronóstico del tiempo; SBR-DR para Bell Canadá, que responde preguntas con respecto a los servicios de la empresa Bell Canadá; SBR-DR para Nobel Prizes & LT-World, que responde preguntas acerca de los premios Nobel y las tecnologías del lenguaje humano; SBR-DR para entorno médico, como su propio nombre lo dice brinda respuestas a consultas en este ámbito.

Como se nota se han realizado investigaciones en diferentes ámbitos como el de la tecnología focalizada en un tipo de sistemas operativo (UNIX), clima, servicios, premios, turismo y médico. La presente propuesta se enfoca en el ámbito de soporte técnico, atendiendo consultas de usuario basadas en texto, que tienen características similares a los tratados en las investigaciones mencionadas, pero que tienen particularidades especiales como son el tratamiento de consultas de inconvenientes de hardware y software de oficina que lo diferencian de estudios anteriores.

En este trabajo se propone una estrategia para aprovechar de forma automatizada, usando el procesamiento del lenguaje natural, la información que se tengan en este ámbito y evitar los esfuerzos de tiempos y costos que amerita obtenerla de forma manual.

Como resultado de esta investigación se desarrollará una herramienta de software BR que permita a través del procesamiento del lenguaje natural identificar lo que el usuario requiere conocer en este ámbito y brindar un tipo de respuesta esperada (TRE).

1.2. Formulación del Problema

La Mesa de ayuda y los problemas técnicos de hardware y software se han presentado desde la aparición del uso masivo de PC's, dentro de una empresa esta labor ha sido siempre una labor de las personas expertas en el tema, que no brindan en muchos casos la solución inmediata al problema, es por ello que tanto los propios usuarios y equipo técnico se apoyan en herramientas de software que les ayuden a resolver algún inconveniente. El estudio e investigación de nuevas tecnologías como son los sistemas RI aportan en algo a mejorar esta situación; pero existen tecnologías especializadas de inteligencia artificial que aportan a este campo como son el PLN y los sistemas de búsqueda respuesta (SBR) que ayudan, como su nombre lo dice, a brindar respuestas a preguntas aportando en gran medida la precisión de las mismas; y mejor aún, si esta investigación se traslada al ámbito restringido en estudio, se busca obtener una comprensión a nivel textual de este tipo de atenciones, y brindar respuestas con un mayor nivel de precisión. Teniendo en contexto lo antes mencionado se puede mencionar que se experimentará para resolver la siguiente pregunta: ¿cómo atender consultas de soporte técnico a nivel de texto?

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1.3. Delimitación de la Investigación

Considerando los antecedentes descritos en anteriores puntos la delimitación espacial de la investigación se circunscribe en el ámbito del Soporte técnico (o también llamado Help desk o mesa de ayuda) considerando las referencias empíricas de las demandas y necesidades de información en esta área específica.

En cuanto a la delimitación temporal se centró en considerar la información recopilada de los últimos seis meses, teniendo en consideración que la información en este ámbito es dinámica, ya que esto se encuentra ligado estrechamente a varios factores como la operativa y crecimiento de las empresas.

1.4. Justificación e Importancia de la Investigación.

El proyecto surge de la necesidad de mostrar las técnicas y métodos de inteligencia artificial, más precisos en el campo del procesamiento del lenguaje natural, que vienen tomando cada vez más relevancia a nivel mundial como son los sistemas de búsquedas de respuestas en ámbitos restringidos que construyen internamente mecanismos computacionales lingüísticos que permiten manejar la estructura del lenguaje y la comprensión del mismo (Ariel Domínguez, 2012).

La justificación que sustenta la ejecución de este proyecto debe sustentarse desde varias aristas dentro de las que se pueden mencionar la científica,

institucional, económica, social y tecnológica, de las cuales se trata a continuación.

1.4.1. Justificación científica.

El procesamiento del Lenguaje natural es una rama de la Inteligencia artificial que permite que la comunicación entre el hombre y la máquina sea más fluida; investiga y formula mecanismos computacionales que permiten esta interacción menos rígida.

En este estudio el procesamiento del lenguaje natural y, siendo más precisos el estudio de los sistemas de búsqueda de respuestas, lleva a un nuevo campo de estudio muy poco difundido en el medio y que servirá de guía para nuevos estudios en este campo.

1.4.2. Justificación social.

En este ámbito la justificación no vendrá hasta que los investigadores los den a conocer y tomen acciones que conduzcan a una educación social, la que permitirá que los logros obtenidos de este tipo de investigaciones sean identificados para luego ser reconocidos, apoyados y explotados en base a las necesidades que plantee el mundo real.

1.4.3. Justificación tecnológica.

Desde el punto de vista tecnológico, es importante mencionar que se estará aplicando inteligencia artificial e ingresando a un campo que viene valorizando su estudio en los últimos años como es el procesamiento del lenguaje natural y que puede servir de base para la construcción de

herramientas tecnológicas modernas que mejoren o complementen las ya existentes.

1.5. Limitaciones de la investigación

Considerando algunos factores que se encontraron en el desarrollo de la investigación se pueden considerar las siguientes:

- La escasa o poca documentación del tema en el idioma español, un gran porcentaje de la documentación del tema tratado se encuentra en inglés u otros idiomas.
- Las herramientas o software de apoyo encontrado son especializados, no muy conocidos y complejos en su uso, tomando en consideración que para hacerlos funcionar es necesario valerse de otras herramientas complementarias igual de complejas.

1.6. Objetivos

Objetivo General

Procesar adecuadamente las consultas del usuario en lenguaje natural basada en texto para mejorar el tipo de respuesta esperada en el ámbito de soporte técnico.

Objetivos Específicos

a) Seleccionar técnicas de procesamiento de lenguaje natural.

b) Diseñar el método de procesamiento de consultas de lenguaje natural.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

c) Implementar en lenguaje de programación el método de procesamiento de consultas.

d) Evaluar los resultados de las pruebas.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO II. MARCO TEÓRICO

La última década se ha caracterizado por grandes cambios dramáticos en la forma de como la internet y la tecnología ha influenciado en la cultura de nuestra sociedad. Durante mucho tiempo se han venido realizando estudios relacionados a la forma de cómo establecer la comunicación hombre máquina teniendo a la inteligencia artificial y más precisa al procesamiento del lenguaje natural como un elemento de estudio que concita gran interés.

En el apartado 2.1 se estará revisando los antecedentes de la investigación que dará acceso a investigaciones anteriores relacionadas y que son materia de estudio, en el apartado 2.2 se estará revisando el Estado del arte que permitirá conocer cómo ha sido tratado el tema, su evolución y las tendencias existentes, en el apartado 2.3 se estará revisando las principales bases teóricas científicas del tema abordado; ya por otro lado en el apartado 2.4 se estará presentando la hipótesis que presenta el enunciado de la posible solución a esta investigación y por último en el apartado 2.5 se estará mostrando las variables intervinientes (dependiente e independiente), estos puntos ayudarán a la mejor comprensión del tema tratado.

2.1. Antecedentes de la Investigación.

Funciones de similitud sobre cadenas de texto: Una comparación basada en la naturaleza de los datos.

En la Universidad pontificia Bolivariana, Iván Amón, y la Universidad Nacional de Colombia, Claudia Jiménez, en el 2012 (Amón & Jiménez,

2012) realizaron un estudio donde trata de la detección de duplicados hace referencia al conflicto que se presenta en los datos cuando una misma entidad del mundo real aparece representada dos o más veces a través de una o varias bases de datos, en registros o tuplas con igual estructura pero sin un identificador único y presentan diferencias en sus valores. Múltiples funciones de similitud han sido desarrolladas para detectar cuáles cadenas son similares mas no idénticas, es decir, cuáles se refieren a una misma entidad. En el presente artículo se compara, mediante una métrica de evaluación llamada discernibilidad, la eficacia de nueve de estas funciones de similitud sobre cadenas de texto (Levenshtein, Brecha Afín, Smith-Waterman, Jaro, Jaro-Winkler, Bi-grams, Tri-grams, Monge-Elkan y SoftTF-IDF) usando para ello seis situaciones problemáticas (introducción de errores ortográficos, uso de abreviaturas, palabras faltantes, introducción de prefijos/sufijos sin valor semántico, reordenamiento de palabras y eliminación/adición de espacios en blanco). Los resultados muestran que algunas funciones de similitud tienden a fallar en ciertas situaciones problemáticas y que ninguna es superior al resto en todas ellas.

La presente investigación basa su importancia en el estudio de la similitud de palabras en el tratamiento de bases de datos, siendo el entendimiento de éstos métodos y técnicas importantes para llevar a cabo la aplicación de los mismos en esta investigación.

Implementación de Técnicas de String Matching y Selección semántica aproximada en un motor de normalización terminológica.

Estíbaliz Parcero Iglesias (2012) desde la Universidad Politécnica de Valencia (España) presentan una investigación acerca de la implementación de técnicas de "String Matching" para un sistema de normalización terminológica.

Este proyecto presenta un sistema de normalización terminológica para el ámbito clínico que busca conseguir un sistema eficaz en base a número de términos correctamente mapeados e integrarlo dentro de un servidor terminológico, donde implementó un motor de comparación de términos basado en técnicas de correspondencia aproximada de cadenas (approximate string matching), ampliamente utilizadas en data mining y de duplicación de datos. Una vez establecida la base teórica se describe la herramienta implementada como una aplicación web e integrada dentro del servidor terminológico. Se le añaden sustanciales mejoras que consiguen aumentar significativamente la eficacia. En conclusión, se consigue una herramienta que centraliza el proceso de normalización terminológica facilitando y ahorrando tiempo al usuario.

Las técnicas de string matching tienen relevancia en esta investigación, ya que permitirá considerar su uso y revisar algoritmos apropiados que permitan cumplir con el objetivo de esta investigación.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Técnicas de procesamiento del lenguaje natural en la recuperación de información.

En la Universidad de Santiago de Compostela (España), Pablo Gamallo Otelo y Marcos García González (2012) presentaron un estudio donde se revisan las diferentes técnicas de recuperación de información, procesamiento del lenguaje natural y análisis de dependencias, evaluando dichas técnicas aplicadas al procesamiento del lenguaje natural. Se estudian, en concreto, métodos de tematización, anotación de categorías morfosintácticas, identificación de nombres propios compuestos y análisis en dependencias. Una evaluación a gran escala con colecciones de documentos en español permitió verificar que la combinación de estas técnicas con otras menos sofisticadas, tales como tokenización y eliminación de palabras gramaticales, que contribuyen a una mejora significativa de la calidad de los sistemas de recuperación.

Para ello se desarrollaron y probaron una herramienta que combina 11 técnicas lingüísticas, tomando como referencia frameworks para el desarrollo que permitió probar la misma.

De la combinación de los 11 sistemas lingüísticos, se llegó a la conclusión que el que mejor resultados brindó es base+deps (sistemas linguísticos) que sobre las demás combinaciones dio los mejores resultados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Las técnicas más sofisticadas de PLN mejoran los sistemas de recuperación del español con respecto a técnicas básicas como la tokenización, simple eliminación de stopwords y el stemming.

De acuerdo a como se muestra el funcionamiento de los sistemas de recuperación de información, se puede decir que, para mejores resultados de éstos, antes de realizar la indexación y expansión de consultas debe pasar por un proceso de lematización, etiquetación y análisis sintáctico.

De manera similar a estudios anteriores, hace referencia en forma detallada al tratamiento y procesamiento de la información imputada que permite captar mejores resultados en los sistemas de recuperación de información, tomándose como referencia en la presente investigación las técnicas computacionales usadas.

Creación automática de sistemas de búsqueda de respuestas en dominios restringidos.

También de la Universidad de Alicante (España) Katia Vila, José-Norberto Mazón y Antonio Fernández (Vila, Mazón, & Ferrández, Creación automática de sistemas de búsqueda de respuestas en dominios restringidos., 2012) presentaron un estudio que dio a conocer una técnica propuesta para la creación automática de sistemas de búsqueda de dominio restringido. Donde se enfrentaron a situaciones como establecer patrones

de pregunta-respuesta de los sistemas de búsqueda de respuesta en ámbito abierto, como también la taxonomía de tipos de respuesta esperada.

Para poder llevar a cabo y cumplir con los objetivos de su estudio se propuso de la revisión de la herramienta Maraqa cuya novedad radica en el uso de técnicas de ingeniería del software, como el desarrollo dirigido por modelos, para automatizar dicho proceso de adaptación a dominios restringidos. Para ello adaptaron un sistema de búsqueda de respuesta de dominio abierto a uno restringido orientado al ámbito agrícola.

Los resultados que obtuvieron indican que la precisión de la búsqueda de respuestas usando Maraqa mejoró a un 58.3% en comparación al 28.8% de mostró AliQan (SBR-DA).

Se puede concluir que en este estudio se presentó la herramienta Maraqa en plataforma Eclipse para realizar la parte técnica de la misma.

La influencia de este estudio en nuestra investigación es importante ya que llama la atención la herramienta Maraqa presentada en este estudio, la cual se investigará su uso a más profundidad, ya que puede aportar al momento de procesar los corpus de nuestro ámbito.

Aplicaciones del procesamiento del lenguaje natural.

En el 2013, Hernández Miriam, Gómez J. de la Universidad de Alicante (España) realizó un estudio sobre las aplicaciones del procesamiento del lenguaje natural, donde se evaluaron diferentes técnicas y formas de

aplicación del procesamiento del lenguaje natural en el ámbito computacional. Se compararon las diferentes formas de implementación de clasificadores de acuerdo a su uso identificando los algoritmos de clasificación más usados, dando como resultado que el algoritmo que mejor respuesta obtuvo fue el SVM (Support vector machine).

Como conclusiones de este estudio se puede mencionar que la clasificación y categorización de textos son los problemas más investigados en procesamiento de lenguaje natural debido a la creciente cantidad de documentos electrónicos existentes en librerías digitales.

Adicionalmente, se comenta la importancia de los datos de entrenamiento para establecer una base de categorización de temas de acuerdo al contexto tratado.

Este estudio tiene influencia en esta investigación ya que aporta detalles del uso de los algoritmos y en especial el SVM para clasificar corpus, que se investigará para evaluar su aplicación en la actualización de la base de conocimiento.

Búsqueda de documentos basada en el uso de índices ontológicos creados por MapReduce

Desde la Universidad Militar Nueva Granada (Bogotá, Colombia 2014) los investigadores Sonia Jaramillo Valbuena y Jorge Mario Londoño presentan un estudio referenciando a los sistemas de búsqueda de

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

documentos basada en el uso de índices ontológicos como principal tema. Este estudio presenta un sistema de búsqueda soportado en un sistema de indexación ontológico.

La técnica presentada utiliza emparejamiento de retículos. El proceso de emparejamiento se realiza entre el retículo podado con el espacio de búsqueda del corpus y el retículo con el espacio de búsqueda de la consulta. Dicho proceso permite realizar un filtrado con los documentos que deben presentarse al usuario. El sistema propuesto fue implementado utilizando el modelo de programación MapReduce. Los resultados experimentales reflejan la eficacia del sistema, al brindar al usuario una mayor correspondencia de los resultados con el dominio de búsqueda.

Además, se evidencian mejoras en el rendimiento y mayor precisión en los resultados mostrados al usuario. La evaluación realizada se incluye al final del artículo.

Al presentar este estudio, las técnicas de emparejamiento se toman como referencia para un probable uso en esta investigación.

MERA: Musical Entities Reconciliation Architecture.

En la Universidad de Oviedo (España), Daniel Fernández Álvarez (Fernández, 2015) en el estudio que le permitió obtener su maestría en Ingeniería WEB presentó un tema importante que permite conocer un poco

más del tratamiento de similitudes y detección de entradas equivalentes en aquellos escenarios en los que no se dispone de identificadores únicos.

Se ha orientado el estudio del caso concreto de las fuentes asociadas al mundo musical, encontrando que no es posible elaborar una lista común de problemas asociados a todas las bases de datos de este tipo. Determinándose que la calidad de datos, idioma, las convenciones de nombres y el tipo de contenido son algunos de los factores que determinan que tipo de estrategias se deberían poner en marcha en cada caso para llevar a cabo satisfactoriamente un proceso de conciliación.

Para aportar una mejora sobre los actuales sistemas de reconocimiento de entidades musicales propone MERA, una arquitectura pensada para el linkado de dos bases de datos capaz de adaptarse a las técnicas y los algoritmos de reconciliación más convenientes en cada caso particular.

En esta propuesta el investigador propone una herramienta basada en Web Semántica usando grafos RDF, en combinación algoritmos de String matching, siendo un aporte importante en este campo y para este estudio.

2.2. Estado del Arte

No cabe duda que el poder de comunicarse a través de la lengua es inherente a los humanos, de acuerdo a los estudios e investigaciones realizadas en el ámbito computacional lingüístico cada vez se está más cerca de que no solo sea la comunicación entre humanos sino también entre

humano – computadora. Actualmente, con el desarrollo del Procesamiento del Lenguaje Natural (PLN), las computadoras pueden ayudar al tratamiento de este conocimiento (Gutiérrez-Artacho, 2014).

2.2.1. Sistemas de Pregunta-Respuesta

Es que es dentro de este contexto que las PLN se encargan de estudiar, diseñar e implementar sistemas computacionales capaces de utilizar y comprender el lenguaje natural posibilitando una comunicación fluida y eficaz entre los seres humanos y las computadoras (Gutiérrez-Artacho, 2014).

Así, la tarea a realizar por los sistemas de Pregunta-Respuesta (sistemas P-R), también conocidos como sistemas de Búsqueda de Respuestas (sistemas BR), y mucho más conocidos por su término inglés Question-Answering Systems (QA systems), se debe clasificar como un tipo de recuperación de información avanzada en el que se parte de una consulta expresada en lenguaje natural y debe devolver no ya un documento que sea relevante (es decir que contenga la respuesta) sino la propia respuesta (normalmente un hecho).

Los SBR se presentan como alternativa a los tradicionales sistemas de Recuperación de información (SRI) que actualmente se usan (Google, Yahoo, IExplorer,etc), que están tratando de incorporar este tipo de sistemas embebidos dentro de sus buscadores como está

pasando actualmente como ayuda a los usuarios que buscan respuesta en el ámbito general (el más notorio y conocido es Google que en ocasiones su motor de búsqueda brinda un tipo de respuesta específica).

Muchos de estos estudios de SBR han evolucionado en tal medida que se vienen generando nuevas variaciones como los SBR multilingüe y translingüe y los ya conocidos monolingües de dominio general y de dominio específico (Gutiérrez-Artacho, 2014).

Historia de los sistemas de los sistemas de pregunta-respuesta

Los primeros sistemas de QA se desarrollaron en los años 60, y prácticamente no eran más que interfaces en lenguaje natural a sistemas expertos construidos para dominios restringidos. Dos de los más famosos fueron BASEBALL y LUNAR. BASEBALL respondía a preguntas sobre la liga de béisbol de USA en el periodo de un año. LUNAR era capaz de responder a preguntas sobre el análisis geológico de las piedras lunares que trajeron las misiones Apollo desde la luna. Ambos sistemas de QA fueron realmente efectivos en sus dominios. En una convención sobre científicos lunares en 1971, LUNAR fue capaz de responder al 90% de las preguntas formuladas por los usuarios, quienes ni si quiera habían sido entrenados por el sistema. A raíz de estos primeros sistemas, se fueron desarrollando otros que mantenían una característica siempre común con los primeros: su núcleo era siempre

una base de datos de conocimiento escrita manualmente por expertos del dominio.

También otros famosísimos sistemas de inteligencia artificial primitivos como SHRLDU y ELIZA incluyeron ciertas habilidades de Pregunta-Respuesta. SHRDLU, desarrollado por Terry Winograd en 1970 (SHRDLU, 1970) era un sistema conversacional (un interfaz de lenguaje natural) que simulaba el comportamiento de un robot para el movimiento de piezas (bloques) en un mundo virtual que era simulado en la pantalla del ordenador. SHRDLU permitía responder a preguntas sobre el estado de este mundo virtual. ELIZA simulaba el comportamiento de un psicólogo computacional. ELIZA era capaz de conversar sobre cualquier tema y tenía una forma muy rudimentaria de responder a las preguntas mediante conversaciones “enlatadas”.

En la década de los 80, con el desarrollo de las teorías de la lingüística computacional empiezan a diseñarse algunos proyectos sobre sistemas de QA más ambiciosos en cuanto a comprensión de textos como el Unix Consultant (UC) que es un sistema capaz de responder a preguntas relativas al sistema operativo UNIX creado sobre una base de conocimiento hecha a mano por expertos del dominio, y que era capaz de acomodar la respuesta según varios tipos de usuario predefinidos (según si era un usuario principiante, experto, etc.). Otro

proyecto interesante fue LILOG, que respondía a preguntas turísticas sobre una ciudad en Alemania.

A partir de 1999, la investigación en sistemas de QA se incorpora a la Text Retrieval Conference1 (TREC-8), formando una competición en la que los sistemas participantes deben responder a preguntas sobre cualquier tema buscando en un corpus de texto. También el Cross Language Evaluation Forum (CLEF) muestra su interés por este tipo de sistemas incluyendo desde 2003 una versión translingual de la competición TREC, en este caso el CLEF-QA. Esta competición está motivada por los siguientes motivos:

- Las respuestas pueden encontrarse en lenguajes diferentes al inglés.
- Un interés creciente en sistemas de QA para lenguajes diferentes al inglés.
- Forzar a la comunidad de QA a diseñar sistemas multilingües reales.
- Comprobar y mejorar la portabilidad de las tecnologías implementadas en los sistemas de QA actuales.

Actualmente existe un creciente interés en encontrar una fusión entre los sistemas de Pregunta-Respuesta y el mundo del World Wide Web. Compañías como Google o Microsoft han empezado a integrar ciertas capacidades de Pregunta-Respuesta en sus motores de

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

búsqueda, y se espera que esta integración sea mucho más importante en un futuro próximo.

Por otro lado, la investigación y el reconocimiento más importante en este campo de los sistemas de Pregunta respuesta es la que se tiene con “IBM Watson” (Chandrasekaran & DiMascio, 2014) que es un sistema que responde a preguntas en lenguaje natural que no usa respuestas preparadas, sino las determina por calificación de confianza de documentos y se basa en el conocimiento adquirido. Comercialmente IBM han generado API’s y herramientas propias que están a disposición de los usuarios, previo pago, para generar herramientas personalizadas a criterio de sus clientes. La limitación que se encuentra con esta herramienta es que actualmente solo se puede usar para el idioma inglés y para poder armar la arquitectura necesaria usa herramientas propias y libres a la vez que se debe tener conocimientos avanzados de desarrollo.

Aplicaciones de los Sistemas de Pregunta-Respuesta

Las aplicaciones de los sistemas de Pregunta-Respuesta son diversas y existe una gran variedad tanto del problema a tratar como de los agentes que intervienen: diferentes tipos de usuario, diferentes formatos de datos, diferentes clases de dominio, etc. Entre los que se pueden mencionar a continuación:

- Atendiendo al tipo de acceso a la información se puede encontrar sistemas de QA que buscan sobre: Datos estructurados (Bases de

Datos), Datos semi-estructurados (XML, estructuras de texto en Bases de Datos) o Texto Libre y la combinación de todos ellos.

- Atendiendo al tipo de colección sobre la que se busca: La Web, Colección de documentos, un texto simple.
- Atendiendo al tipo de dominio: Dominio libre, Dominios restringidos (alta precisión)
- Atendiendo al modo en el que se presenta la información: Texto, Imágenes, Datos hablados, Vídeo, etc.
- Atendiendo al tipo de usuario: Usuarios casuales, noveles sin perfil, acceso general, usuarios expertos, con perfil definido y con acceso a información específica.

2.2.2. Sistemas de Búsqueda de Respuesta de Dominio Restringido (SBR-DR)

Como ya se había mencionado en un punto anterior, los sistemas de BR de dominio restringido (SBR-DR) es un sistema diseñado para adaptarse de forma óptima a un área concreta. Dentro de estos sistemas pueden tratarse diferentes tipos de preguntas, siendo el ámbito de aplicabilidad el que determine qué tipo de preguntas resulten más interesantes (Vila, Mazón, & Ferrández, Creación automática de sistemas de búsqueda de respuestas en dominios restringidos., 2012) .

En la actualidad los SBR-DR abarcan varios temas entre ellos se han encontrado sistemas para los ámbitos: turísticos, médico, de pronóstico del

tiempo, académico entre otros interesantes (Vila, Mazón, & Ferrández, Creación automática de sistemas de búsqueda de respuestas en dominios restringidos., 2012), de los cuales se hacen mención a continuación:

- EXTRANS: Para dar respuesta a preguntas arbitrarias sobre archivos de documentación UNIX.
- WEBCOOP: Que da respuesta a varios aspectos de dominios turísticos parcialmente restringidos ya que incluye determinados aspectos de historia, seguridad, salud, inmigración y ecología.
- SBR-DR para pronóstico del tiempo: Robot doméstico que contesta preguntas sobre el pronóstico del tiempo.
- SBR-DR para Bell Canadá: Para responder preguntas con respecto a los servicios de la empresa Bell Canadá.
- SBR-DR para Nobel Prizes & LT-World: Para responder preguntas acerca de los premios Nobel y las tecnologías del lenguaje humano.
- SBR-DR para entorno médico: Para dar respuesta para responder consultas en el ámbito médico.

2.2.3. String Matching en PLN

Se han llevado diversos estudios en lo que se hacen uso de esta técnica en las que se hacen diversos experimentos para estandarización de terminologías haciendo uso de motores de comparación de términos

basado en correspondencia aproximada de cadenas, ampliamente utilizadas en data mining y duplicación de datos (Parcero, 2012).

En estudios más recientes se han estudiado casos concretos de datos asociadas al mundo musical, encontrando que no es posible homologar información de este tipo si no se considera la calidad de datos, el idioma y las convenciones de nombres que determinan las estrategias que se deberían poner en marcha para afrontar esta problemática (Fernández, 2015).

Estos estudios brindan beneficios a este estudio, puesto que las aplicaciones de estas técnicas dan luces que es posible hacer con ellas y clarifican el camino para poder incluirlas dentro de un sistema de búsqueda de respuesta.

2.3. Bases Teórico – Científicas

2.3.1. Mesa de ayuda

1. ¿Qué es una Mesa de ayuda?

Una Mesa de ayuda es un área de una empresa que responde a las preguntas técnicas de los usuarios. La mayoría de las grandes empresas tienen Mesas de ayuda para responder a las preguntas de los usuarios. Las respuestas a estas preguntas pueden ser entregadas por teléfono, correo, chat o boletines. (Wooten & Wooten, 2001)

2. Mesa de ayuda en las empresas.

En muchas áreas de una empresa, las diversas formas de servicio, de la mesa de ayuda, proporcionan soluciones a los usuarios. En los servicios de mesa de ayuda convencionales, grupos de expertos humanos que difieren en conocimientos y experiencia tratan de resolver los problemas de los clientes. Sus funciones se determinan en función de su capacidad para resolver problemas y el grado de dificultad que plantea el problema. Por lo tanto, para proporcionar un servicio de mesa de ayuda de alta calidad, la disponibilidad de expertos de alto nivel es crucial. Sin embargo, el número de este tipo de expertos de alto nivel es limitado.

Es en base a estos conceptos y apuntes mencionados, se puede comentar adicionalmente que para proporcionar un mejor y preciso servicio se deriva la Gestión de niveles del servicio (SLM): El objetivo general de este proceso es garantizar que se cumplen los niveles de provisión de los servicios de TI, tanto existentes como futuros, de acuerdo a los objetivos acordados. (Bon, y otros, 2008), que deben ponerse en práctica haciendo mérito a las mejores prácticas del ITIL (Information Technologies Infraestructure Library).

2.3.2. Inteligencia Artificial

1. Definición de Inteligencia artificial

Como lo menciona (Quevedo, Rosique, Ruiz, & Aldeguer, 1999) en su libro Fundamentos de la Inteligencia Artificial, antes de definir el término Inteligencia Artificial se deberá definir qué se entiende por inteligencia. La Real Academia de la Lengua Española define la inteligencia como "Potencia intelectual: facultad de conocer, de entender o comprender". Como ya se tiene una definición formal de este término ahora se pregunta lo mismo pero aplicado a las máquinas. Partiendo de esta premisa, cuando se menciona que un aparato electrónico es inteligente se tendría que preguntar qué conocimiento o comprensión tiene dicho aparato del proceso que realiza. Como se puede observar, este término es muy ambiguo y, como se verá más adelante, se han aducido argumentos a favor y en contra de la inteligencia en las máquinas.

Se dará una definición de inteligencia artificial que se piensa es más cercana a la realidad. La propuso Marvin Misky, uno de los pioneros de la IA., y dice así: **"La Inteligencia Artificial es la ciencia de construir máquinas para que hagan cosas que, si las hicieran los humanos, requerirían inteligencia"**. A partir de esta definición ya se tiene acotado el campo de estudio.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Se piensa entonces en la IA como la ciencia que incorpora conocimientos a los procesos o actividades para que éstos tengan éxito. Un ejemplo es el ajedrez, es impensable que un ordenador evalúe todas las posibles jugadas del ajedrez. En vez de esto, se incorpora conocimiento en el proceso de búsqueda de la mejor jugada en forma de jugadas predefinidas o procedimientos de evaluación "inteligentes".

2. Resumen histórico de la IA.

La historia de la Inteligencia artificial se puede resumir en el siguiente cuadro (Hardy, 2006), que muestra la evolución de la misma desde sus inicios desde el siglo XIX a la actualidad.

Cuadro 1 : Resumen histórico de la Inteligencia Artificial, explicando los datos mas relevantes de las investigaciones.

834:Ancestro	El matemático Charles Babbage (1792-1871) define el concepto de máquina universal, ancestro del computador moderno y propone los planos.
1936: Máquinas de Turing	El matemático inglés Alan M. Turing (1912-1954) define una máquina abstracta, la “Máquina de Turing”, que sirve de base a la noción del algoritmo y a la definición de la clase de problemas decidibles. Turing dedicó lo principal de sus trabajos a la formalización de la teoría de los autómatas y a la noción de la calculabilidad.
1943: Primer computador	El catalizador que condujo al verdadero nacimiento de la IA fue la aparición del primer computador, el ENIAC: una máquina electrónica de programa grabado.
1950: Test de Turing	Turing propone en un artículo clásico: “Can a machine think?”, la definición de una experiencia que permitiría calificar a una máquina de inteligente. La experiencia consiste en que un computador y algún voluntario humano se oculten a la vista de algún (perspicaz) interrogador. Este último tiene que tratar de decidir cuál de los dos es el computador y cual el ser humano,

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

	mediante el simple procedimiento de plantear preguntas a cada uno de ellos. Si en el curso de una serie de test semejantes, la interrogadora es incapaz de identificar la naturaleza de su interlocutor, se considera que el computador ha superado la prueba.
1956: Noción de listas.	J.Mc Carthy se da la tarea de construir un lenguaje de programación adaptable a las necesidades de manipulación de conocimientos y de la reproducción de razonamientos basados en la noción de listas.
1959: General Problem Solver	Después de las investigaciones de A. Newell y H. Simón sobre los mecanismos de razonamientos, surge el GPS: General Problem Solver, basado en el principio del "análisis de los fines y de los medios". El interés de GPS es el de haber sido el primero a formalizar el razonamiento humano. Su meta era investigar sobre la actividad intelectual y sobre los mecanismos puestos en juego durante la resolución de problemas, más que en la eficacia.
Años sesenta: heurística	Los años sesenta marcan la verdadera puesta en marcha de la IA, con algunos resultados significativos: Enumeración

	inteligente de soluciones a través de reglas optativas o heurísticas (heurística: arte de inventar, P. Larousse, 1995).
1966: Dificultades	Reconocimiento por parte de los investigadores, de la dificultad del reconocimiento de la palabra y de la traducción de las lenguas. Un trabajo de fondo deberá por lo tanto comprenderse sobre el análisis y la definición de las estructuras de lengua sobre la base de los trabajos de N. Chomsky.
Años sesenta: ¡boom!	Los años setenta corresponden a una explosión de trabajos que permitieron establecer las bases de la IA, en cuanto a la representación de los conocimientos, del razonamiento, de los sistemas expertos, de la compresión del lenguaje natural y de la robótica avanzada.
1970: Primer sistema experto	Aparición de **Dendral**, el primero de los sistemas expertos, en la Universidad de Stanford. Dendral efectúa el trabajo de un químico que reconstituye la fórmula desarrollada de un componente orgánico a partir de la fórmula bruta y de los resultados de su espectrografía de masa.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1975: Programación en lógica de primer orden.	Aparece PROLOG de la universidad de Aix Marseille (Francia) y marca los comienzos de una verdadera programación basada en la lógica de primer orden. Este lenguaje conoció un tal éxito que fue adoptado como el lenguaje de base para el proyecto japonés de los computadores de quinta generación.
1976: Medicina.	MYCIN, sistema experto en diagnósticos de infecciones bacterianas de la sangre para la ayuda de la antibioterapia de Schortliffe. Sus principales características son, por una parte, la separación de los conocimientos del mecanismo de razonamiento y el diálogo en lenguaje casi-natural y por otra parte, la asistencia para los ajustes de las bases hacia la industrialización.
Los años ochenta: IA y economía.	Los años ochenta son aquellos de la entrada de la IA en la vida económica. Con realizaciones prácticas importantes en diferentes áreas y, paralelamente, de un crecimiento notable de los esfuerzos de investigación a través de proyectos muy ambiciosos en la mayoría de los países industrializados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1981: Computadores de quinta generación.	Lanzamiento en Japón del proyecto del computador de quinta generación. El objetivo anunciado para el proyecto es el desarrollo de tecnologías de la IA en la realización de un nuevo tipo de computadores que resolverán problemas en lugar de ejecutar los algoritmos, que efectuarán razonamientos en vez de solo cálculos y ofrecería a sus usuarios interfaces naturales: Lenguaje, gráfica, palabra.
Años noventa: Comunicación hombre máquina.	Los años noventa marcan la entrada de la IA en las aplicaciones vinculadas a la comunicación hombre-máquina con interfaces inteligentes, sistema multi-agentes y la IA distribuida.
Futuro: ¿computación cuántica?	Qubit: bit cuántico = "la lógica de un bit es uno u otro, mientras que el qubit entraña el concepto de ambos a la vez, sean cuatro respuestas posibles + ¡el estado de una partícula se determina a través de la asignación de una probabilidad!" ¿Qué deparará el futuro?

Fuente: Hardy, 2006

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3. Tecnologías relacionadas a la Inteligencia Artificial

Existen diferentes tipos de tecnologías que forman parte de lo que se conoce como inteligencia artificial.

Por un lado, existen los sistemas expertos, programas de cómputo que tienen conocimientos específicos sobre un tema. Estos conocimientos son dados por expertos humanos en un área en particular y colocados dentro del programa junto con reglas y heurísticas.

Estos programas tienden a enfocarse directamente en el tema sobre el cual se especializan y no tienen la facultad de aprender de su experiencia.

La estadística también es usada en la inteligencia artificial, principalmente cuando el sistema se enfrenta a la incertidumbre producida por la falta de información. En estos casos, el sistema es capaz de tomar una decisión aun teniendo datos incompletos.

El software de inteligencia artificial puede tener mecanismos que le permitan el aprendizaje, como es el caso de las redes neuronales. Cuando se requiera un sistema de control sencillo, se puede usar una máquina de estados finitos.

Algunas veces, es necesario trabajar con datos de los cuales no se conocen los valores exactos; en estos casos, se utiliza

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

la lógica difusa. Por su parte, los algoritmos genéticos pueden llegar a soluciones para un problema en particular.

4. Aplicaciones de la Inteligencia Artificial[1]

El ámbito de aplicación de la inteligencia artificial incluye:

- Tratamiento de Lenguajes Naturales: Capacidad de Traducción, Órdenes a un Sistema Operativo, Conversación Hombre−Máquina, etc.
- Sistemas Expertos: Sistemas que se les implementa experiencia para conseguir deducciones cercanas a la realidad.
- Robótica: Navegación de Robots Móviles, Control de Brazos móviles, ensamblaje de piezas, etc.
- Problemas de Percepción: Visión y Habla, reconocimiento de voz, obtención de fallos por medio de la visión, diagnósticos médicos, etc.
- Aprendizaje: Modelización de conductas para su implante en computadoras.

5. Procesamiento del Leguaje Natural

El procesamiento del lenguaje natural (PLN), área de investigación en continuo desarrollo, se aplica en la actualidad en diferentes actividades como son la traducción automática, sistemas de recuperación de información, elaboración automática de

[1] (Biblioteca Universidad Católica Santo Toribio de Mogrovejo, 2006, pág. 2)

resúmenes, interfaces en lenguaje natural, etc. Si bien en los últimos años se han realizado avances espectaculares, los fundamentos teóricos del PLN se encuentran todavía en estado de desarrollo (Sosa, 1997).

Aun siendo evidente que los obstáculos a superar en el estudio del tratamiento del lenguaje son considerables, los resultados obtenidos y la evolución en los últimos años sitúan al PLN en posición para liderar una nueva dimensión en las aplicaciones informáticas del futuro: los medios de comunicación del usuario con el ordenador pueden ser más flexibles y el acceso a la información almacenada más eficiente (Allen, 1995).

No obstante, la complejidad implícita en el tratamiento del lenguaje comporta limitaciones en los resultados y, por tanto, aplicaciones en áreas de conocimiento concretas y con un uso restringido del lenguaje.

I. Revisión histórica

Las primeras aplicaciones del PLN se dieron durante el período de 1940-1960, teniendo como interés fundamental la traducción automática. Los experimentos en este sector, basados en la substitución de palabra por palabra, obtuvieron resultados rudimentarios.

Surgió por tanto la necesidad de resolver ambigüedades sintácticas y semánticas, y asimismo la consideración de información contextual. La carencia de un orden de la estructura oracional en algunas lenguas, y la dificultad para obtener una representación tanto sintáctica como semántica, fueron los problemas más relevantes. Afrontándolos se dio paso a una concepción más realista del lenguaje en la que era necesario contemplar las transformaciones que se producen en la estructura de la frase durante el proceso de traducción.

En los años sesenta los intereses se desplazan hacia la comprensión del lenguaje. La mayor parte del trabajo realizado en este período se centró en técnicas de análisis sintáctico.

Hacia los setenta la influencia de los trabajos en inteligencia artificial fue decisiva, centrando su interés en la representación del significado. Como resultado se construyó el primer sistema de preguntas-respuestas basado en lenguaje natural.

De esta época es Eliza, que reproducía las habilidades conversacionales de un psicólogo. Para ello recogía patrones de información de las respuestas del cliente y elaboraba preguntas que simulaban una entrevista.

Entre los años 70 y 80, ya superados los primeros experimentos, se hacen intentos de construir programas más fiables. Aparecen numerosas gramáticas orientadas a un tratamiento computacional, y

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

experimenta notable crecimiento la tendencia hacia la programación lógica.

En Europa surgen intereses en la elaboración de programas para la traducción automática. Se crea el proyecto de investigación Eurotra, que tenía como finalidad la traducción multilingüe. En Japón aparecen equipos dedicados a la creación de productos de traducción para su distribución comercial.

Los últimos años se caracterizan por la incorporación de técnicas estadísticas y se desarrollan formalismos adecuados para el tratamiento de la información léxica. Se introducen nuevas técnicas de representación del conocimiento cercanas a la inteligencia artificial, y las técnicas de procesamiento utilizadas por investigadores procedentes del área de la lingüística e informática son cada vez más próximas. Surgen así mismo intereses en la aplicación de estos avances en sistemas de recuperación de información con el objetivo de mejorar los resultados en consultas a texto completo (Allen, 1995).

II. Concepto de procesamiento del lenguaje natural

El PLN se concibe como el reconocimiento y utilización de la información expresada en lenguaje humano a través del uso de sistemas informáticos.

En su estudio intervienen diferentes disciplinas tales como lingüística, ingeniería informática, filosofía, matemáticas y psicología.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Debido a las diferentes áreas del conocimiento que participan, la aproximación al lenguaje en esta perspectiva es también estudiada desde la llamada ciencia cognitiva.

Tanto desde un enfoque computacional como lingüístico se utilizan técnicas de inteligencia artificial: Modelos de representación del conocimiento y de razonamiento, Lenguajes de programación declarativos, Algoritmos de búsqueda y estructuras de datos.

Se investiga cómo el lenguaje puede ser utilizado para cumplir diferentes tareas y la manera de modelar el conocimiento.

En los siguientes párrafos se presenta una introducción a las técnicas que se aplican para el tratamiento del lenguaje natural. Generalmente la bibliografía sobre el tema se caracteriza por su estilo técnico y, dado su componente interdisciplinar, se presenta como una materia de difícil comprensión para los legos en el tema.

En vista a conocer estas técnicas de representación y procesamiento, es necesario tener en cuenta una doble dimensión: se trata por una parte de un problema de representación lingüística, y por otra de un problema de tratamiento mediante recursos informáticos.

El uso de técnicas computacionales procedentes especialmente de la inteligencia artificial no aportaría soluciones adecuadas sin una concepción profunda del fenómeno lingüístico. Por otra parte, las

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

gramáticas utilizadas para el tratamiento del lenguaje han evolucionado hacia modelos más adecuados para un tratamiento computacional.

III. Cuatro niveles de análisis

El estudio del lenguaje natural se estructura normalmente en 4 niveles de análisis (Bach, 1989): Morfológico, Sintáctico, Semántico y Pragmático.

Análisis Morfológico

Su función consiste en detectar la relación que se establece entre las unidades mínimas que forman una palabra, como puede ser el reconocimiento de sufijos o prefijos. Este nivel de análisis mantiene una estrecha relación con el léxico.

El léxico es el conjunto de información sobre cada palabra que el sistema utiliza para el procesamiento. Las palabras que forman parte del diccionario están representadas por una entrada léxica, y en caso de que ésta tenga más de un significado o diferentes categorías gramaticales, tendrá asignada diferentes entradas.

En el léxico se incluye la información morfológica, la categoría gramatical, irregularidades sintácticas y representación del significado (Chierchia, 1990).

Normalmente el léxico sólo contiene la raíz de las palabras con formas regulares, siendo el analizador morfológico el que se encarga de

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

determinar si el género, número o flexión que componen el resto de la palabra son adecuados.

Análisis Sintáctico

Tiene como función etiquetar cada uno de los componentes sintácticos que aparecen en la oración y analizar cómo las palabras se combinan para formar construcciones gramaticalmente correctas. El resultado de este proceso consiste en generar la estructura correspondiente a las categorías sintácticas formadas por cada una de las unidades léxicas que aparecen en la oración (Chierchia, 1990).

Las gramáticas, tal como se muestra en la siguiente figura, están formadas por un conjunto de reglas:

O --> SN, SV

SN --> Det, N

SN --> Nombre Propio

SV --> V, SN

SV --> V

SP --> Preposición, SN

SN = sintagma nominal

SV = sintagma verbal

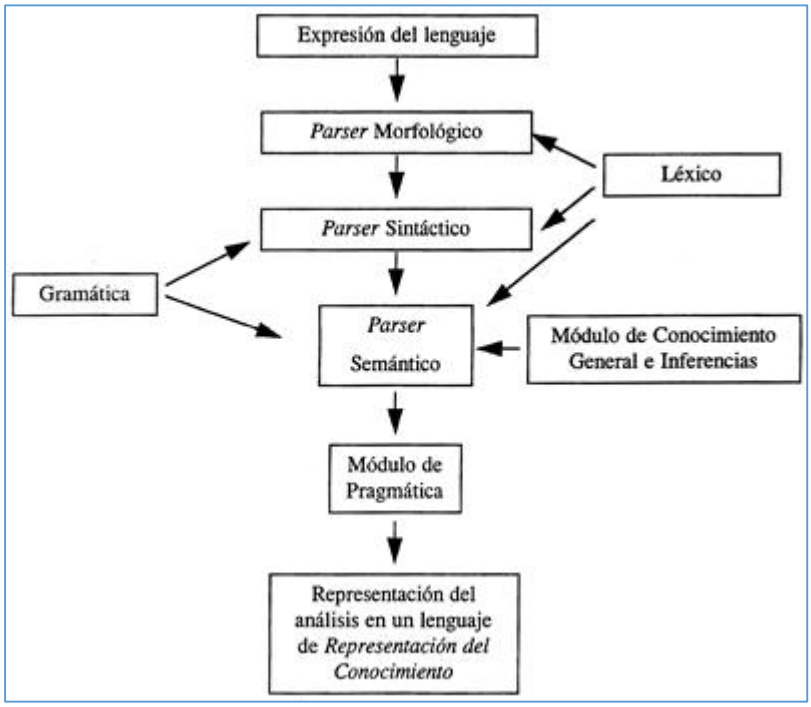
Det = determinante

Ejemplo de una gramática simple: las reglas tienen como función la composición de estructuras.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

El resultado del análisis se puede expresar en forma arbórea. Los árboles son formas gráficas utilizadas para expresar la estructura de la oración, consistentes en nodos etiquetados (O, SN, SV.) conectados por ramas:

Figura 1: Ejemplo árboles de representación del análisis.



Fuente: (Sosa, 1997)

Análisis Semántico

En muchas aplicaciones del PLN los objetivos del análisis apuntan hacia el procesamiento del significado. En los últimos años las técnicas de procesamiento sintáctico han experimentado avances significativos, resolviendo los problemas fundamentales.

Sin embargo, las técnicas de representación del significado no han obtenido los resultados deseados, y numerosas cuestiones continúan sin encontrar soluciones satisfactorias (McEnery, 1992).

Definir qué es el significado no es una tarea sencilla, y puede dar lugar a diversas interpretaciones. A efectos funcionales, para facilitar el procesamiento, la modularidad es una de las propiedades más deseables. Haciendo uso de esta concepción modular es posible distinguir entre significado independiente y significado dependiente del contexto (Chierchia, 1990).

El primero, tratado por la semántica, hace referencia al significado que las palabras tienen por sí mismas sin considerar el significado adquirido según el uso en una determinada circunstancia. La semántica, por tanto, hace referencia a las condiciones de verdad de la frase, ignorando la influencia del contexto o las intenciones del hablante. Por otra parte, el componente significativo de una frase asociada a las circunstancias en que ésta se da, es estudiado por la pragmática y conocido como significado dependiente del contexto.

Atendiendo al desarrollo en el proceso de interpretación semántica, es posible optar entre múltiples pautas para su organización, tal como se determinan en los siguientes párrafos.

En referencia a la estructura semántica que se va a generar, puede interesar que exista una simetría respecto a la estructura sintáctica, o

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

por el contrario que no se dé tal correspondencia entre ellas. En el primer caso, a partir del árbol generado por el análisis sintáctico se genera una estructura arbórea con las mismas características, sobre la cual se realizará el análisis semántico. En el segundo caso, en la estructura generada por la sintaxis se produce un curso de transformaciones sobre las cuales se genera la representación semántica.

Cada una de las dos opciones anteriores puede implementarse de forma secuencial o paralela. En la interpretación secuencial, después de haber finalizado la fase de análisis sintáctico, se genera el análisis semántico. En cambio, desde un procedimiento en paralelo, el proceso de análisis semántico no necesita esperar a que el analizador sintáctico haya acabado toda su tarea, sino que puede ir realizando el análisis de cada constituyente cuando éste ha sido tratado en el proceso sintáctico.

Finalmente, en combinación con cada una de las opciones anteriores, se puede escoger un modelo en el que exista una correspondencia entre reglas sintácticas y semánticas o, contrariamente, se puede optar por un modelo que no cumpla tal requisito. En caso afirmativo, para cada regla sintáctica existirá una regla semántica correspondiente.

El significado es representado por formalismos conocidos por el nombre de knowledge representation. El léxico proporciona el

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

componente semántico de cada palabra en un formalismo concreto, y el analizador semántico lo procesa para obtener una representación del significado de la frase (Reichgelt, 1991).

Análisis Pragmático

Añade información adicional al análisis del significado de la frase en función del contexto donde aparece. Se trata de uno de los niveles de análisis más complejos, la finalidad del cual es incorporar al análisis semántico la aportación significativa que pueden hacer los participantes, la evolución del discurso o información presupuesta.

Incorpora asimismo información sobre las relaciones que se dan entre los hechos que forman el contexto y entre diferentes entidades.

Analizadores Sintácticos

Parsing es el término con el que se denomina el proceso de análisis sintáctico realizado en PLN. Consiste en un conjunto de operaciones algorítmicas necesarias para la comprensión sintáctica de una frase, teniendo como input una frase y como output una representación arbórea del resultado (McEnery, 1992).

Las operaciones de análisis que se realizan son agrupamientos parciales de los signos léxicos en unidades superiores. Por ejemplo, un determinante combinado con un nombre será tratado por una regla que permita la creación de un sintagma nominal. Esta unidad sintagmática se utilizará en otras reglas para formar unidades superiores. En el

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

momento que se haya analizado la frase se va a tener conocimiento de todas las unidades léxicas que aparecen y la relación establecida entre ellas. El resultado de este análisis puede ser utilizado para realizar un análisis semántico (Bach, 1989).

El parser o analizador produce inferencias a partir de la información consultada en el módulo de gramática. Sus algoritmos son considerados procedimientos de búsqueda que dirigen el proceso y elaboran la representación de la estructura de la frase. La función de un parser es, por tanto, identificar los elementos de la oración y especificar la relaciones entre ellos (Meya & Huber, 1986).

Procedimientos de análisis

En función de la organización del análisis y la combinación de reglas se diferencia entre analizadores descendentes y analizadores ascendentes.

Analizadores descendentes

Conocidos también con el nombre de top-down, se caracterizan por comenzar el proceso de análisis por las reglas más generales: se considera la representación del análisis en forma arbórea, en la parte superior (top) aparecen las reglas más generales, y a medida que desciende se procesan las más específicas, llegando finalmente a las unidades léxicas formadas por palabras (Reichgelt, 1991).

En modo top-down el programa comienza por reglas generales como O --> SN, SV (indica que una oración está formada por un sintagma nominal seguido de un sintagma verbal). La siguiente regla a aplicar sería del tipo:

SN --> Det, Nombre (indica que un sintagma nominal está formado por un determinante seguido de un nombre). En el proceso de parsing de la frase el cliente compra un libro se aplicaría las siguientes reglas según este orden:

O --> SN, SV

SN --> Det, N

Det--> el

N --> cliente

SV --> V, SN

V --> compra

SN --> Det, N

Det--> un

N --> libro

Analizadores Ascendentes

Conocidos también como bottom-up, aplican las reglas en orden inverso al modelo anterior: el análisis comienza por la parte inferior del árbol (bottom), y se forman unidades complejas ascendiendo hacia la parte superior (up) (Meya & Huber, 1986).

El primer paso es el reconocimiento de las categorías de las palabras:

Det--> el

N --> cliente

Para formar posteriormente unidades sintagmáticas en forma ascendente.

SN --> Det, N

Los estados de análisis son generados por la asignación de una palabra a su categoría léxica correspondiente, o bien reemplazando la parte derecha de una regla por su parte izquierda.

Algunas aplicaciones del PLN

A pesar de que hoy en día sus aplicaciones son limitadas, a medida que se incorporen nuevos avances el uso de técnicas de PLN puede comportar la integración en numerosas actividades.

Son abundantes los estudios realizados en el campo del PLN en los últimos años y numerosas las ramas de investigación surgidas. Entre las principales tareas en este campo destacan: Traducción automática (Hutchins y Somers, 1992), Recuperación de información (Baeza-Yates y Ribeiro-Neto, 1999), Reconocimiento del habla (Junqua y Haton, 1995), Resúmenes automáticos (Mani, 1999), Interfaces en lenguaje natural a bases de datos (Androutsopoulos et

al., 1995), Detección de autoría (Juola, 2007), Análisis de sentimientos (Pang y Lee, 2008), Búsqueda de respuestas (Pasca, 2003), Extracción de información (Cardie, 1997), Desambiguación del sentido de las palabras (Agirre y Edmonds, 2006), Implicación textual (Dagan et al., 2006), Clasificación de textos (Sebastiani, 2002), Reconocimiento de entidades (Palmer y Day, 1997).

Consecuentemente, las publicaciones y foros sobre el tema han sido muy numerosos. Una de las revistas más conocidas en este campo es Computational Linguistics, la revista oficial de The Association for Computational Linguistics, que desde 1974 está dedicada exclusivamente a investigaciones sobre el diseño y análisis de sistemas de PLN. Ofrece información sobre aspectos computacionales de investigación en lenguaje, lingüística y psicología del procesamiento del lenguaje.

IV. Técnicas de Análisis del lenguaje

Como mencionan (F.J. & J.L., 2012) en su presentación sobre Inteligencia Artificial, específicamente sobre las técnicas del Procesamiento de Lenguaje Natural; las distintas fases y problemáticas del análisis del lenguaje se afrontan principalmente con las siguientes técnicas: **Técnicas Lingüísticas Formales** que son aquellas que se basan en el desarrollo de reglas estructurales que se aplican en las

fases de análisis del lenguaje y las **Técnicas Probabilísticas** que se basan en el estudio a un conjunto de textos de referencia (corpus) de características de tipo probabilístico asociadas a las distintas fases de análisis del lenguaje.

A continuación, se da un alcance de los dos tipos de técnicas usadas en el procesamiento del lenguaje natural, estas son:

- **Técnicas Lingüísticas formales.**

  Como lo menciona (Pedraza-Jimenez & Vallez, 2007) son técnicas y reglas que codifican en forma explícita el conocimiento lingüístico (Sanderson, 2000). Los documentos son analizados a partir de los diferentes niveles lingüísticos por herramientas lingüísticas que incorporan al texto las anotaciones propias de cada nivel.

  En este punto ya no se entrará en detalles porque de ello se habló en el punto 2.3 Base Teórico-Científicas que da alcances del tratamiento de estas técnicas.

  De manera general se dará un ejemplo descriptivo que permitirá tener una mejor visión de este tema (Pedraza-Jimenez & Vallez, 2007):

  El análisis morfológico es ejecutado por los etiquetadores (taggers) que asignan a cada palabra su categoría gramatical a partir de los rasgos morfológicos identificados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Después de identificar y analizar las palabras que forman un texto, el siguiente paso consiste en ver cómo éstas se relacionan y combinan entre sí para formar unidades superiores, los sintagmas y las frases. Por tanto, se trata de realizar el análisis sintáctico del texto. En este punto se aplican gramáticas (parsers) que son formalismos descriptivos del lenguaje que tienen por objetivo fijar la estructura sintáctica del texto. Las técnicas empleadas para aplicar y construir las gramáticas son muy variadas y dependen del objetivo con el que se realiza el análisis sintáctico. En el caso de la recuperación de la información acostumbra a aplicarse un análisis superficial, donde se identifican únicamente las estructuras más significativas como frases nominales, sintagmas verbales y preposicionales, entidades, etc. Este nivel de análisis suele utilizarse para optimizar recursos y no ralentizar el tiempo de respuesta de los sistemas.

A partir de la estructura sintáctica del texto, el siguiente objetivo es obtener el significado de las frases que lo componen. Se trata de conseguir la representación semántica de las frases, a partir de los elementos que la forman.

- **Técnicas Probabilísticas:**

  Como lo menciona (Martínez-Barco, Vicedo, Saquete, & Tomás, 2009) en su tratado acerca de Sistemas de Pregunta

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

respuesta, existen varias técnicas que son utilizadas en las tres fases de estos sistemas (Pre procesamiento, Recuperación y Extracción), los cuales se mencionarán a continuación para entrar en contexto:

- **Pre procesamiento de la consulta:**

  Aquí se mencionará algunas técnicas primitivas que se usan en muchos casos en combinación con otras para cumplir con los objetivos del estudio, entre ellas se tienen:

  a. **Bag of words:**

  Técnica consistente en considerar toda la consulta como una lista de palabras sueltas que directamente se introducen en el recuperador de información sin contemplar ningún tipo de extensión ni orden de importancia entre ellas.

  b. **Stemming:**

  Técnica computacional consistente en reducir una palabra flexionada o derivada a su stem o forma raíz, es decir, la parte de la palabra que es invariante a todas sus formas flexionadas eliminando los sufijos.

  c. **Lematización:**

  Técnica computacional para determinar el lema de una palabra. Este proceso ya implica determinar la categoría

gramatical de la palabra por lo que se requiere de una gramática de la lengua y de un diccionario.

d. **Eliminación de palabras de paso o stopwords:**

Se llama “palabra de paso” o stopword a aquellas palabras que se eliminan sistemáticamente previo a un proceso de análisis de un texto. La consideración de ser palabra de paso o no viene dada por la utilidad que pueda tener esa palabra dentro de un uso o contexto determinado. En el caso de los sistemas de Pregunta-Respuesta se consideran típicamente palabras de paso los artículos y algunas preposiciones.

e. **Análisis sintáctico superficial.**

Los sistemas de análisis sintáctico proporcionan datos importantes al análisis de la pregunta. Mediante un análisis sintáctico superficial (o chunking) se pueden detectar constituyentes básicos necesarios para la búsqueda de información como pueden ser los grupos nominales y verbales, aunque no llega a determinar sus constituyentes internos ni el rol que ocupan en la oración. La salida obtenida por este tipo de herramientas sería una lista plana de constituyentes básica.

f. **Análisis sintáctico de dependencias.**

El análisis de dependencias no sólo detecta los constituyentes básicos sino también las dependencias y

relaciones entre ellos. Son determinantes para conseguir obtener el rol sintáctico de cada constituyente en la oración.

g. **Desambiguación del sentido de las palabras.**

La necesidad de la determinación de las características semánticas de algunos términos de la consulta para permitir su correcta clasificación implica el uso de desambiguadores que proporcionan el sentido exacto del término contra una ontología previamente establecida.

h. **Obtención de representaciones semánticas.**

Mediante el uso de estructuras complejas como la formas lógicas se puede llegar a obtener una representación completa sintáctico-semántica que representa un cierto nivel de organización mental de la pregunta (Zajac, 2001).

i. **Sistema de clasificación de la pregunta.**

Se trata de la clasificación en un conjunto limitado de clases. La forma más básica distingue entre clases de acuerdo con la partícula interrogativa. Clasificaciones más complejas relacionan el tipo de pregunta con el tipo de respuesta esperado creando a su vez subclases de preguntas.

j. **Obtención del foco de la pregunta.**

En algunas ocasiones, el tipo de la pregunta es tan genérico que no aporta nada sobre el tipo de información que

se está buscando. Es el caso de las preguntas tipo WHAT básicas: What is the largest city in Germany? El problema se soluciona con la definición de foco de la pregunta, que es una palabra o conjunto de palabras que define la pregunta y la desambigua indicando lo que se está buscando, en este caso, “largest city”. Al conocer el tipo de pregunta y conocer su foco resulta mucho más fácil identificar el tipo de información que se espera como respuesta.

k. **Extracción de los términos clave de la pregunta.**

Por último, es necesario establecer cuáles son los términos clave que se van a buscar. Los sistemas más simples se limitan a extraer las palabras de la pregunta eliminando las palabras de paso.

- **Recuperación de la información**:

La recuperación de información de la información relevante es el segundo módulo que se debe considerar para cumplir con el objetivo del estudio. El proceso de recuperación de información varía mucho dependiendo del tipo de acceso a los datos contenedores de la respuesta:

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

a. **Recuperación de información sobre datos estructurados:**

En este caso la recuperación de información se transforma en un acceso a la base de datos mediante su lenguaje de consulta. Para ello se deberán establecer mecanismos de traducción de la pregunta al lenguaje de consulta (normalmente SQL). Para relacionar los términos de la pregunta con el esquema de la base de datos es necesario establecer un mapeo previo de la base de datos con la ontología. Puramente hablando, el esquema de la base de datos debería constituir en sí la propia ontología del sistema de QA. De esta forma, tras analizar la pregunta se obtendrían todos los parámetros necesarios para construir la consulta a la BD, ya que cada uno de sus términos se corresponderá con un objeto de la BD. En este sentido hay algunas propuestas basadas en la transformación de formas lógicas a SQL.

b. **Recuperación de información sobre datos no estructurados o semi estructurados:**

En este caso, al no existir una estructura previa, no se puede realizar un mapeo directo de los términos de la pregunta en la colección documental. La información

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

susceptible de contener la respuesta debe ser analizada y relacionada con la pregunta.

- **Extracción de la respuesta:**

El proceso de extracción de la respuesta comprende la localización de la respuesta en los fragmentos relevantes. La dificultad de esta tarea depende del tipo de respuesta esperada. Por ejemplo, la respuesta a un Who? puede ser tan simple como encontrar una entidad de tipo persona existente en el fragmento relevante. La fase de extracción puede subdividirse en las siguientes tareas:

a. **Análisis de los pasajes relevantes:**

Se usan generalmente técnicas de análisis superficial (shallow parsing), con especial énfasis en el reconocimiento de entidades con nombre. En algunas ocasiones se han utilizado gramáticas específicas para cada uno de los tipos de respuesta a buscar. La complejidad del análisis vendrá determinada por el modelo de proyección usado.

b. **Extracción de la respuesta:**

Dependen totalmente del modelo de proyección que se haya elegido, pero en su forma más básica se usan técnicas simples de pattern matching donde se establecen diferentes

medidas según el tipo de atributo (coincidencia de términos, densidad de términos relevantes, dispersión, etc.) y que luego se combinan.

c. **Validación de la respuesta:**

En algunos sistemas, una vez localizada la respuesta se intenta a través de algún tipo de razonamiento demostrar que realmente responde a la pregunta. El problema de la validación se puede definir como: “Dada una pregunta Q y un candidato a respuesta A, decidir si A es la respuesta correcta para Q” (Magnini, y otros, 2005).

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO III. Marco Metodológico

3.1. Tipo de estudio

3.1.1. Tipo de Investigación

El tipo de investigación seleccionada es la Explicativa aplicada; se dice que es explicativa ya que se busca de la necesidad de estudiar este campo de conocimiento de la inteligencia artificial, dando a conocer el impacto que tendrá en el ámbito de estudio. Se menciona que es aplicada ya que se presentará una alternativa práctica de solución al problema planteado, reforzándolo con la teoría relacionada.

3.1.2. Diseño de la investigación

El diseño de nuestra investigación es “Experimental”, y como se escribió en el punto anterior; teóricamente se menciona que permitirá presentar una propuesta de solución al problema planteado en la presente tesis en relación a las variables planteadas.

3.2. Población y Muestra

El presente caso de estudio hace referencia a consultas de soporte técnico por lo que:

Población: Será la cantidad de consultas de texto diarias que ocurren, que es alrededor de 200 al día, de acuerdo a los datos obtenidos por la experiencia de especialistas en este campo; sin embargo los investigadores

en sistemas inteligentes establecen un rango de datos para los experimentos (Vicedo González - 2002, Mesones Barrón – 2006, Gamallo Otero & García González- 2012) y basados en esa función de las investigadas y realizadas.

Muestra: Se determina en función a la población, con la siguiente fórmula:

$$n = \frac{NZ^2PQ}{NE^2 + Z^2PQ}$$

Dónde:

n: es el tamaño de la muestra;

Z: es el nivel de confianza;

P: es la variabilidad positiva;

Q: es la variabilidad negativa;

N: es el tamaño de la población;

E: es la precisión o el error.

Entonces se calcula la muestra:

$$n = \frac{200 * (1.28)^2 * (0.5) * (0.5)}{200 * (0.2)^2 + (1.28)^2 * (0.5) * (0.5)}$$

$$n = \frac{81.92}{8{,}4096}$$

$$n = \ 9{,}741248097$$

$$\boldsymbol{n = 10\ consultas}$$

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.3. Hipótesis

Utilizando las técnicas del procesamiento del lenguaje natural se podrá resolver las consultas de usuario a nivel de texto.

3.4. Operacionalización

Desde el punto de vista de la operacionalización de las variables se muestra el siguiente cuadro que da a conocer los indicadores que permitirán medir esta investigación.

Cuadro 2 : Operacionalización de las variables. Presenta las variables a medir.

ÍTEM	DIMENSIÓN	INDICADOR	PREGUNTA	TIPO	UN. MED	CATEGORÍA
1	Tiempo	Comprensión de consultas	¿Cuánto tiempo demora la comprensión de consultas?	Cuantitativo Continuo	ms.	[0 - 10] [11 - 20] … [91 - 100]
2	Precisión	Porcentaje de resultados correctos.	¿Qué porcentaje de resultados brindados por la herramienta son correctos?	Cuantitativo Discreto.	%	[0 - 10] [11 - 20] … [91 - 100]

Fuente: Propia del tema de investigación.

3.5. Métodos, técnicas e instrumentos de recolección de datos

Para poder encarar este proyecto de investigación se ha considerado aplicar como métodos la experimentación y observación.

3.6. Plan de análisis estadístico de datos

El análisis de los datos recopilados, conforme a lo mencionado en el punto anterior se estará haciendo uso de la media aritmética y la varianza;

la misma información será procesada y analizada con herramientas informáticas que es el resultado de esta investigación, y, como soporte adicional se hará uso de MS Excel.

3.7. Análisis Estadístico e Interpretación de los datos

El análisis de los datos recopilados, conforme a lo mencionado en el punto anterior se estará haciendo uso de la media aritmética y la varianza; la misma información será procesada y analizada con el uso de MS Excel.

3.8. Criterios éticos

Los progresos en la investigación que aportan conocimiento conllevan a muchos beneficios; los mismos deben realizarse teniendo en consideración los criterios éticos de nivel profesional que corresponde.

Es en este sentido que la presente investigación contempla dentro de su desarrollo estos criterios, los cuales se han sabido respetar, como el de la propiedad intelectual. Las investigaciones y autores, de las mismas, han sido nombrados e incluidos en nuestra bibliografía dándoles el crédito de los aportes brindados a la presente investigación.

3.9. Criterios de rigor científico

Con el afán de cumplir el objetivo de esta investigación se ha considerado ciertos criterios de rigor científico que permiten hacer fiable los hallazgos presentados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

En relación a estos criterios se puede mencionar como criterios clave y diferenciador el de “credibilidad” y “confirmabilidad”. Con respecto al primero de ellos se ha seguido la pista de los autores de investigaciones anteriores basados en la revisión de la documentación pertinente la que a su vez permiten dar claridad a esta; en el caso del segundo criterio (confirmabilidad), cabe mencionar que tiene relación directa con la primera ya que se puede llegar a la fuente de las investigaciones realizadas, sabiendo que esta investigación toma como base en muchos de los casos la evidencia documentaria encontrada de los autores citados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

CAPÍTULO IV. ANALISIS E INTERPRETACIÓN DE LOS RESULTADOS

4.1. Resultados.

Las medidas de evaluación empleadas miden el rendimiento de la herramienta SIPLENAST (**Si**stema **P**rocesamiento **Len**guaje **Na**tural para **S**oporte **T**écnico) en función a la rapidez de devolución de una respuesta (tiempo) y la información que se pide (precisión).

Para evaluar los resultados del proyecto se muestra los resultados de un conjunto de pruebas, los mismos que fueron recopilados de corpus del ámbito de soporte técnico y almacenados en la base de datos para representar el caso de estudio.

A continuación se muestra los resultados obtenidos con la herramienta en función del tiempo en **milisegundos (ms)** que demora el procesamiento del texto ingresado en la consulta con respecto a la respuesta brindada.

Cuadro 3 : Resultado de Tiempos transcurrido por consulta.

Ítem	Pregunta almacenada	Texto de consulta realizada	Hora inicio	Hora fin	Tiempo transcurrido (UM = ms)
1	¿Por qué mi PC disminuyó su rendimiento?	¿Por qué mi equipo disminuye su rendimiento?	1 Dic 2016 06:43:22 GMT 1480833802328	1 Dic 2016 06:43:22 GMT 1480833802442	114
		¿Mi PC disminuyó su rendimiento, a que se debe?	1 Dic 2016 06:46:29 GMT 1480833989462	1 Dic 2016 06:46:29 GMT 1480833989604	142
		¿Por qué mi computadora disminuyó su rendimiento?	1 Dic 2016 06:49:47 GMT 1480834187013	1 Dic 2016 06:49:47 GMT 1480834187158	145

2	¿Cuáles son las Causas comunes de los atascos de papel?	¿Por qué se me atasca el papel?	1 Dic 2016 06:51:44 GMT 1480834304995	1 Dic 2016 06:51:45 GMT 1480834305088	93
		¿Cuáles son las causas de atasco de papel?	1 Dic 2016 06:52:53 GMT 1480834373500	1 Dic 2016 06:52:53 GMT 1480834373588	88
		Causas comunes de los atascos de papel	1 Dic 2016 06:53:51 GMT 1480834431802	1 Dic 2016 06:53:51 GMT 1480834431881	79
3	La impresora toma varias hojas	Porqué la impresora toma varias hojas	1 Dic. 2016 06:55:16 GMT 1480834516282	1 Dic 2016 06:55:16 GMT 1480834516410	128
		La impresora toma varias hojas	1 Dic 2016 06:56:11 GMT 1480834571185	1 Dic 2016 06:56:11 GMT 1480834571240	55
		¿Por qué me toma varias hojas?	1 Dic 2016 07:01:18 GMT 1480834878047	1 Dic 2016 07:01:18 GMT 1480834878171	124
4	¿Cómo soluciono si Mi pc no reconoce mi disco duro externo?	¿Mi PC no reconoce mi disco externo que puedo hacer?	1 Dic 2016 07:04:14 GMT 1480835054252	1 Dic 2016 07:04:14 GMT 1480835054379	127
		Solución cuando no reconoce mi disco externo	1 Dic 2016 07:06:13 GMT 1480835173814	1 Dic 2016 07:06:13 GMT 1480835173956	142
		¿Cómo soluciono si Mi pc no reconoce mi disco duro externo?	1 Dic 2016 07:07:37 GMT 1480835257259	1 Dic 2016 07:07:37 GMT 1480835257324	65
5	Aparecen pantallas de error en el Windows, los programas se tildan y se cierran	Me aparecen pantallas de error y mis programas se cierran	1 Dic 2016 07:11:35 GMT 1480835495642	1 Dic 2016 07:11:35 GMT 1480835495781	139
		Aparecen pantallas de error en el Windows, los programas se tildan y se cierran	1 Dic 2016 07:12:38 GMT 1480835558801	1 Dic 2016 07:12:38 GMT 1480835558852	51
		Sale error en Windows y se cierran mis programas	1 Dic 2016 07:13:47 GMT 1480835627769	1 Dic 2016 07:13:47 GMT 1480835627861	92
6	¿Qué hago si tengo mucha información y es muy importante?	Qué puedo hacer si tengo mucha información importante?	1 Dic 2016 07:43:03 GMT 1480837383536	1 Dic 2016 07:43:03 GMT 1480837383746	210

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

		¿Qué hago si tengo mucha información y es muy importante?	1 Dic 2016 07:44:05 GMT 1480837445641	1 Dic 2016 07:44:05 GMT 1480837445835	194
		Tengo mucha información importante, qué puedo hacer?	1 Dic 2016 07:45:11 GMT 1480837511608	1 Dic 2016 07:45:11 GMT 1480837511738	130
7	¿Cómo diagnosticar un disco duro externo?	¿Cómo diagnostico un disco externo?	1 Dic 2016 07:47:58 GMT 1480837678125	1 Dic 2016 07:47:58 GMT 1480837678247	122
		¿Cómo diagnosticar un disco duro externo?	1 Dic 2016 07:48:54 GMT 1480837734830	1 Dic 2016 07:48:54 GMT 1480837734888	58
		Cómo diagnostico mi disco duro externo si está fallando?	1 Dic 2016 07:52:52 GMT 1480837972211	1 Dic 2016 07:52:52 GMT 1480837972300	89
8	¿Cómo conectar varias computadoras a una impresora mediante una red?	¿Cómo conecto mi computadora a una impresora de red?	1 Dic 2016 07:57:01 GMT 1480838221050	1 Dic 2016 07:57:01 GMT 1480838221243	193
		Conectar varias computadoras a una impresora de red	1 Dic 2016 07:58:57 GMT 1480838337649	1 Dic 2016 07:58:57 GMT 1480838337785	136
		¿Cómo conectar varias computadoras a una impresora mediante una red?	1 Dic 2016 07:58:01 GMT 1480838281384	1 Dic 2016 07:58:01 GMT 1480838281441	57
9	¿Cómo soluciono el atasco de papel dentro de la impresora?	¿Cómo soluciono el atasco de papel dentro de la impresora?	1 Dic 2016 07:59:51 GMT 1480838391585	1 Dic 2016 07:59:51 GMT 1480838391667	82
		Se me atasco el papel en la impresora, que puedo hacer?	1 Dic 2016 08:01:08 GMT 1480838468079	1 Dic 2016 08:01:08 GMT 1480838468177	98
		¿Cómo solucionar el atasco del papel?	1 Dic 2016 08:02:20 GMT 1480838540792	1 Dic 2016 08:02:20 GMT 1480838540912	120
10	La impresora no imprime a doble cara o lo hace de forma incorrecta.	¿Qué puedo hacer si no me imprime a doble cara?	1 Dic 2016 08:03:41 GMT 1480838621206	1 Dic 2016 08:03:41 GMT 1480838621301	114
		La impresora no me imprime a doble cara, que puedo hacer?	1 Dic 2016 08:05:06 GMT 1480838706437	Dec 2016 08:05:06 GMT 1480838706579	142

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

	La impresora no imprime a doble cara o lo hace de forma incorrecta.	1 Dic 2016 08:05:57 GMT 1480838757365	1 Dic 2016 08:05:57 GMT 1480838757424	59

Fuente: Propia del tema de investigación.

Como se puede observar en el cuadro 9 se ha extraído información de la base de datos estructurada entrenada por corpus del ámbito de Soporte técnico, de los cuales se han seleccionado un grupo de 10 preguntas. En la primera columna se puede ver el dato almacenado de la pregunta de referencia y en la siguiente columna se muestra el texto en lenguaje natural con las que se realizó la consulta, se realizaron tres variaciones de la pregunta para poder tomar el tiempo que toma SIPLENAST en responder. Lo que se puede notar es dos cosas: la primera, con respecto a la pregunta almacenada, que cuanto más extensa es la pregunta mayor es el tiempo que demora el motor de consulta; la segunda es que cuanto más se asemeja el texto ingresado a la pregunta original el tiempo de procesamiento es más rápido. Del grupo seleccionado que se probó, en general, se obtuvo un promedio de 112.93 ms, como se muestra en el cuadro 4.

Cuadro 4 : Resultado de Tiempos Promedio de consultas.

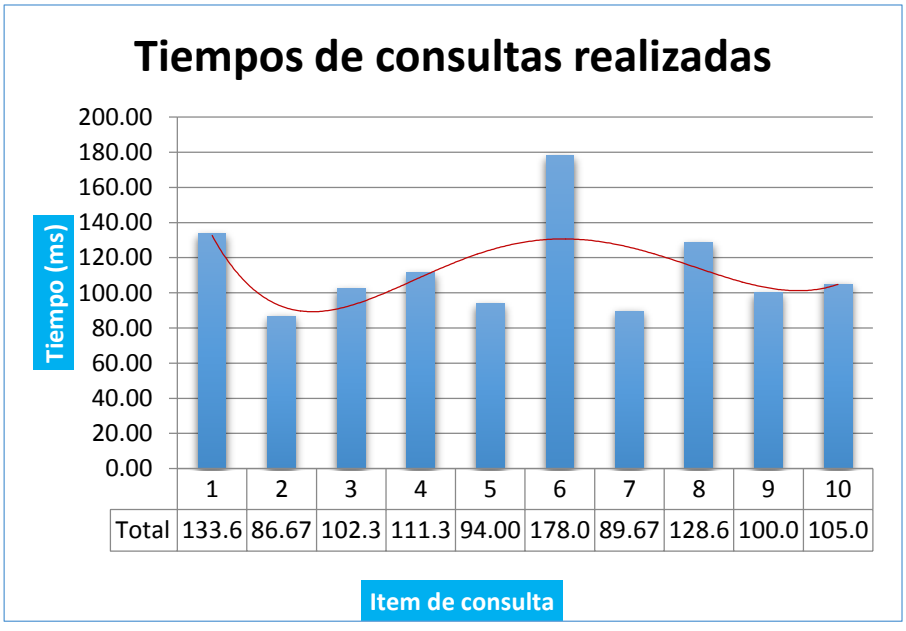
Ítem	Pregunta almacenada	Promedio (ms)
1	¿Por qué mi PC disminuyó su rendimiento?	133,67
2	¿Cuáles son las Causas comunes de los atascos de papel?	86,67
3	La impresora toma varias hojas	102,33

4	¿Cómo soluciono si Mi pc no reconoce mi disco duro externo?	111,33
5	Aparecen pantallas de error en el Windows, los programas se tildan y se cierran	94,00
6	¿Qué hago si tengo mucha información y es muy importante?	178,00
7	¿Cómo diagnosticar un disco duro externo?	89,67
8	¿Cómo conectar varias computadoras a una impresora mediante una red?	128,67
9	¿Cómo soluciono el atasco de papel dentro de la impresora?	100,00
10	La impresora no imprime a doble cara o lo hace de forma incorrecta.	105,00
	Total general	**112,93**

Fuente: Propia del tema de investigación.

En el siguiente gráfico se puede notar la diferencia que existe entre la diferentes consultas y como fluctúa los tiempos en base a cada consulta realizada.

Figura 2 : Gráfico de Tiempos transcurrido por consulta



Fuente: Propia del tema de investigación.

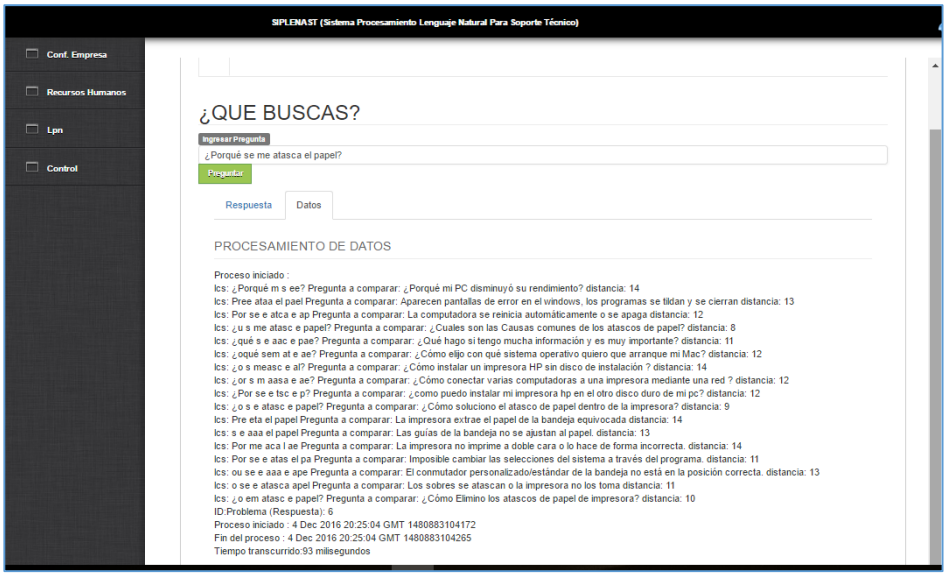
Para poder tener a disposición y de manera visual los resultados de las búsquedas del motor de búsqueda de SIPLENAST se ha implementado una pestaña que muestra de manera sencilla y visual el proceso de selección de la mejor respuesta para la consulta ingresada, la misma que se muestra a continuación:

Figura 3 : Pantalla que muestra una búsqueda y su resultado.

Fuente: Propia del tema de investigación.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 4 : Pantalla que muestra el proceso de selección de la búsqueda y su resultado.



Fuente: Propia del tema de investigación.

La misma muestra seleccionada fue sometida a la prueba de precisión, como lo muestra el siguiente cuadro.

Cuadro 5 : Resultado de precisión de consultas.

Ítem	Pregunta almacenada	Texto de consulta realizada	Encontró
1	¿Por qué mi PC disminuyó su rendimiento?	Porqué mi equipo disminuye su rendimiento?	SI
		¿Mi PC disminuyó su rendimiento, a que se debe?	SI
		Mi pc no rinde a que se debe?	NO
2	¿Cuáles son las Causas comunes de los atascos de papel?	¿Por qué se me atasca el papel?	SI
		¿Cuáles son las causas de atasco de papel?	SI

		Causas comunes de los atascos de papel	SI
3	La impresora toma varias hojas	Porqué la impresora toma varias hojas	SI
		La impresora toma varias hojas	SI
		Porqué me toma varias hojas la impresora?	NO
4	¿Cómo soluciono si Mi pc no reconoce mi disco duro externo?	Mi PC no reconoce mi disco externo que puedo hacer?	SI
		Solución cuando no reconoce mi disco externo	SI
		¿Cómo soluciono si Mi pc no reconoce mi disco duro externo?	SI
5	Aparecen pantallas de error en el Windows, los programas se tildan y se cierran	Me aparecen pantallas de error y mis programas se cierran	SI
		Aparecen pantallas de error en el Windows, los programas se tildan y se cierran	SI
		Sale error en Windows y se cierran mis programas	SI
6	¿Qué hago si tengo mucha información y es muy importante?	Qué puedo hacer si tengo mucha información importante?	SI
		¿Qué hago si tengo mucha información y es muy importante?	SI
		Tengo mucha información importante, qué puedo hacer?	SI
7	¿Cómo diagnosticar un disco duro externo?	Cómo diagnostico un disco externo?	SI
		¿Cómo diagnosticar un disco duro externo?	SI

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

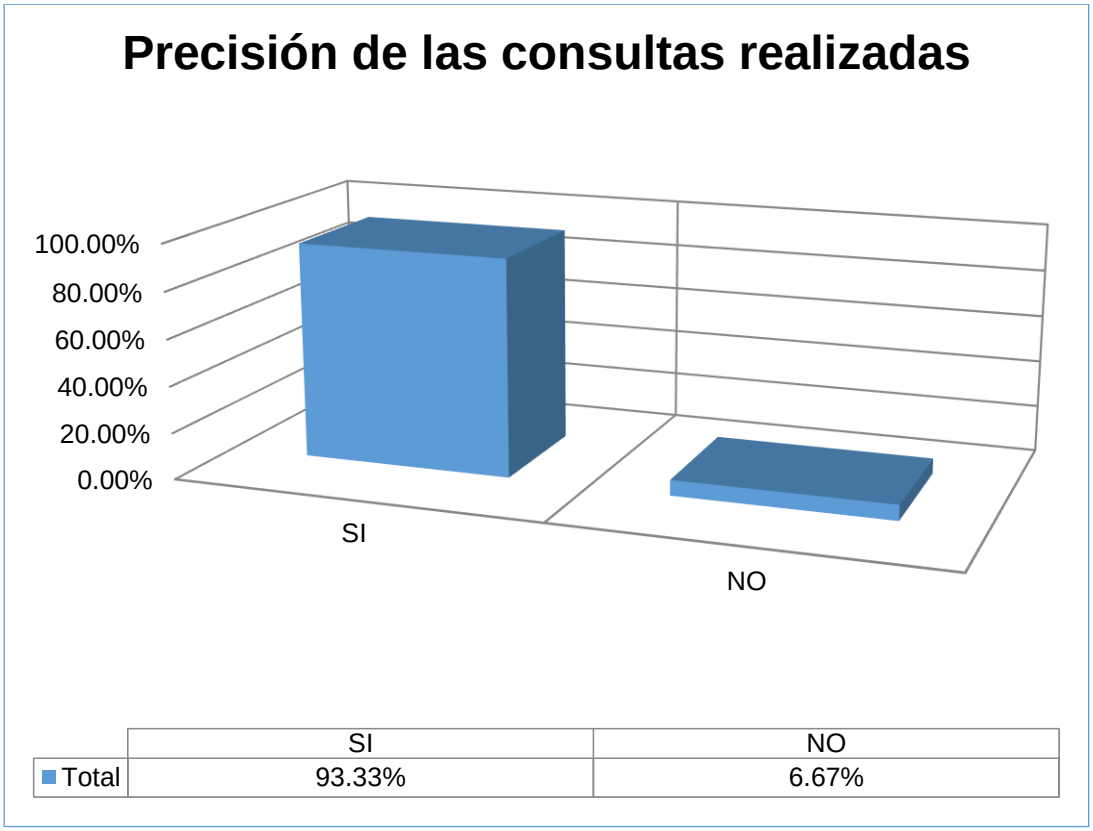
		Cómo diagnostico mi disco duro externo si está fallando?	SI
8	¿Cómo conectar varias computadoras a una impresora mediante una red ?	¿Cómo conecto mi computadora a una impresora de red?	SI
		Conectar varias computadoras a una impresora de red	SI
		¿Cómo conectar varias computadoras a una impresora mediante una red?	SI
9	¿Cómo soluciono el atasco de papel dentro de la impresora?	¿Cómo soluciono el atasco de papel dentro de la impresora?	SI
		Se me atasco el papel en la impresora, que puedo hacer?	SI
		¿Cómo solucionar el atasco del papel?	SI
10	La impresora no imprime a doble cara o lo hace de forma incorrecta.	¿Qué puedo hacer si no me imprime a doble cara?	SI
		La impresora no me imprime a doble cara, que puedo hacer?	SI
		La impresora no imprime a doble cara o lo hace de forma incorrecta.	SI

Fuente: Propia del tema de investigación.

Esto supone haber encontrado respuesta a varias consultas de texto (en esta caso tres) para cada pregunta de la muestra, lo que da un resultado muy alentador con respecto a cómo se ha desarrollado el motor de búsqueda ya que arroja un 93.33% de asertividad en brindar la respuesta

requerida contra un 6.67% de fallo, que más va ligada a como se plantea la consulta.

Figura 5 : Gráfico de porcentajes de precisión de consultas realizadas.



Fuente: Propia del tema de investigación.

Se puede decir en definitiva que se ha logrado desarrollar una herramienta que, de manera sencilla, cumple con el objetivo de la investigación y será un aporte significativo para los usuarios ya que va a permitir resolver las consultas de manera más precisa.

4.2. Discusión de resultados.

En este apartado se estará discutiendo aquellos aspectos relevantes del estudio. El objetivo es procesar adecuadamente las consultas del usuario en lenguaje natural basada en texto para mejorar el tipo de respuesta esperada en el ámbito de Soporte técnico. Por esto es que en el presente punto se discutirá los resultados reportados en cuadros de punto anterior en convergencia con la revisión de literatura relacionada y bases teóricas.

4.2.1. En relación a un antecedente.-

Del análisis de los resultados de este estudio se puede decir que se ha logrado cumplir con el objetivo de estudio, considerando que la herramienta SIPLENAST ha logrado cubrir con precisión en un 93.33% con solo un 6.67% de fallo con un tiempo de respuesta promedio de 112.93 ms (milisegundos) por respuesta. Adicionalmente se puede comentar que la precisión de los resultados va de la mano con el planteamiento de la consulta.

Si se quiere comparar los resultados mostrados con otro estudios, se va a tener que comentar que si bien es cierto que se ha encontrado estudios que usan técnicas similares como los estudios de “Implementación de String matching y selección semántica” de (Parcero, 2012) y la “Búsqueda de documentos basada en índices ontológicos” de (Jaramillo & Londoño, 2014), no se podrá hacer

comparación de resultados por diferir en el uso de estas técnicas y el ámbito de estudio.

4.2.2. En relación a las bases teóricas.-

En este aspecto se va a comentar que, después de haber revisado estudios realizados en esta rama de estudio (PLN) e identificar la técnica de Levenshtein como la técnica a usar para desarrollar la herramienta SIPLENAST, se puede mencionar que se ha mejorado el uso de esta técnica surgiendo una evolución en el uso de este algoritmo aplicado a un motor de búsqueda, esto se sustenta en que en estudios anteriores el algoritmo de Levenshtein se usa para realizar comparación simples de cadenas de caracteres, en esta investigación se hizo una modificación al motor de búsqueda complementándolo con el algoritmo LCS (Longest common subsequence) para que soporte la preselección y comparación de frases largas con un costo mínimo de tiempo y precisión, es por eso que se considera que esta investigación es un gran aporte a este campo de estudio y en especial a nuestra región y a nivel nacional, ya que en Perú es un campo muy poco difundido y estudiado.

CAPÍTULO V. PROPUESTA DE INVESTIGACIÓN

En este capítulo se describirá de manera detallada el proceso que llevó al desarrollo de la herramienta resultante de ésta caso de estudio, con el fin de cumplir con el objetivo general de esta investigación. Se considera pertinente hacer mención la técnica seleccionada para el desarrollo de esta herramienta y el diseño de la misma, se detallará y explicará de la manera más didáctica posible el detalle técnico y código utilizado para el desarrollo de la herramienta, así como también se estará evaluando los resultados que se obtuvieron.

5.1. Selección de la técnica de procesamiento de lenguaje natural.

Lo que se pretende con este trabajo es la realización de una herramienta que asista al usuario de manera más precisa en la atención de consultas sobre el ámbito de soporte técnico con el uso de técnicas de procesamiento de lenguaje natural.

Para poder llegar a seleccionar una técnica que ayude a cumplir con el objetivo de este proyecto se revisaron varias técnicas, muchas de las cuales son propias para tipos de estudio más especializados y relacionados a la lingüística computacional y ontológica (Web 2.0).

Las técnicas implementadas en este motor de búsqueda que más se ajusta a la necesidad son aquellas basadas en la descomposición del término en unidades básicas, este proceso se denomina tokenización (de esto se obtiene un conjunto de tokens o lo que es lo mismo Q-gramas), estas se combinan con la técnica de Stopwords (eliminación de palabras de paso) y el Chunking

(Análisis sintáctico superficial). Este grupo de técnicas son denominadas String Matching y hacen uso de operaciones de teoría de conjuntos aplicadas al conjunto de tokens generados por el proceso de tokenización para calcular la similitud entre cadenas las mismas que se usan en el motor de comparación.

Entre las opciones disponibles de estas técnicas, se han tomado como referencia las siguientes: Levenshtein, SFS, Jaro, Jaccard entre otros. Por no ser un objetivo de esta investigación no se ha llegado a evaluar de manera detallada y realizar la comparación del rendimiento de cada una de estas técnicas, pero si se ha revisado trabajos que tuvieron como objetivo realizar esta labor, tales como los trabajos realizados por Iván Amón - Universidad Pontificia Bolivariana -y Claudia Jiménez –Universidad Nacional de Colombia- (Iván & Claudia, 2010), y el trabajo realizado por (Ruiz, Juárez, Cervantes, & Trueba, 2015) – Universidad Autónoma del Estado de México. Estos estudios evaluaron el rendimiento de estas técnicas y muestran el siguiente resultado:

Cuadro 6 : Resultado de métrica de cadenas.

Técnica	Max F1	AvgFree	
SFS	0,528	0,036	
Jaccard	0,567	0,402	
L2 JaroWinkler	0,746	0,770	
SoftTFIDF	0,685	0,782	
Jaro-Winkler	0,648	0,703	
NaiveAvgOverlap	0,697	0,731	
AvgOverlap	0,701	0,736	
Jaro	0,728	0,789	Recortado
Scaled Levenshtein	0,851	0,093	Recortado
Levenshtein	**0,865**	**0,925**	**Recortado**

Fuente: (Ruiz, Juárez, Cervantes, & Trueba, 2015)

Como se nota en los resultados mostrados en cuadro 7 la técnica con mayor precisión es la de Levenshtein, concluyendo que las técnicas de Jaro y las de Levenshtein muestran eficiencia en el cálculo de las métricas. Al ser la técnica de Levenshtein la más precisa, es la que se aplicó en el desarrollo del motor de búsqueda de la herramienta.

Básicamente este algoritmo considera que las operaciones de borrado e inserción tienen un costo de 1 mientras que la de sustitución tiene un costo de 2 ya que la considera como compuesta por una operación de borrado más una de inserción, para que sea más claro el funcionamiento del algoritmo se da un ejemplo (Rodríguez, 2012):

Ejemplo: "intención" y "ejecución"

```
I N T E *  N C I Ó N
|  |  |  |  |  |  | |  |  |
* E J E C U C I Ó N
b  s  s     i  s => (b: borrado, s: sustitución e i: inserción)
```

Distancia: 5 (si cada operación cuesta 1)

¿Por qué aparecen unos * en medio de la palabra?

Para encontrar el menor número de operaciones de edición es importante que los strings estén alineados de la forma más conveniente. Se brinda un ejemplo para que quede más claro: "mente" y "sutilmente":

```
* * * * * M E N T E
|  |  |  |  |  |  |  |  |  |
S U T I L M E N T E
i  i  i  i  i
```

Distancia: 5 (si cada operación cuesta 1)

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Si se alinean las palabras para que coincidan las partes iguales se están ahorrando 5 operaciones.

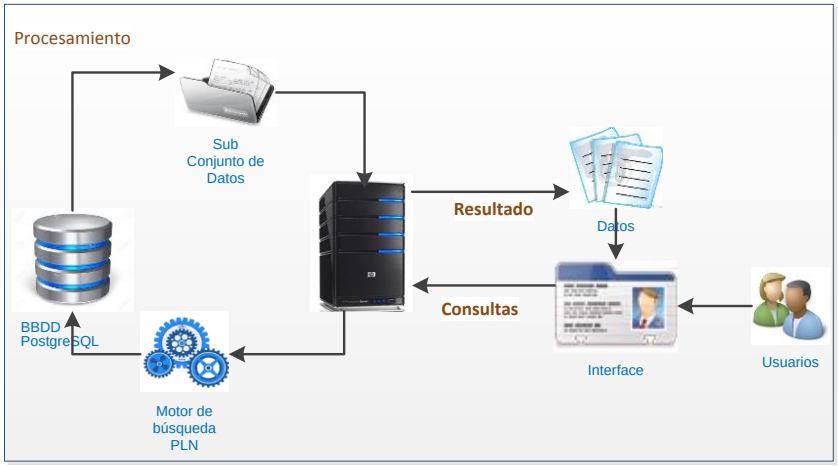
5.2. Diseñar el método de procesamiento de consultas de lenguaje natural.

En este punto se está dando a conocer como se ha realizado la herramienta, para ello se describirá desde la arquitectura de la aplicación, las tecnologías empleadas como también la metodología de desarrollo utilizado para el análisis y diseño de la herramienta software.

5.2.1. Arquitectura de la aplicación.

En el presente apartado se presenta la arquitectura de la solución que da una visión general de cómo funciona la herramienta desarrollada que sustenta esta investigación.

Figura 6 : Arquitectura de la solución *Fuente: Diseño propio.*



Como muestra la figura 6, lo que hará la herramienta de manera general será solicitar una entrada a manera de consulta en lenguaje natural la cual será procesada por el motor de búsqueda, la cual estará realizando una búsqueda en nuestro corpus estructurado que devolverá un sub conjunto de datos de donde se extraerá los datos solicitados como resultado de esta interacción.

5.2.2. Tecnologías empleadas.

Dentro de las tecnologías empleadas que sirvieron de soporte para el diseño y desarrollo de nuestra herramienta, se tienen:

- **Netbeans IDE 8.0.2:**

  Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Se seleccionó por ser amigable, además de contar con un número importante de módulos para extenderlo.

- **Lenguajes de Desarrollo:**

  Si bien es cierto que se hizo uso de Netbeans como IDE para el desarrollo, vale mencionar que el lenguaje usado para la lógica del negocio y Servlets es JAVA. Adicionalmente como complemento también se hizo uso de la tecnología JSP (JavaServer Pages), JQUERY (biblioteca de JavaScript), HTML y CSS (Cascading Style Sheets) para la capa de presentación de la herramienta.

- **Apache Tomcat:**

  Es un contenedor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

- **PostgreSQL:**

  Es una herramienta de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL.

  Todas estas tecnologías fueron usadas teniendo en consideración dos aspectos: el tecnológico, ya que son multiplataforma, permiten desarrollar herramientas robustas y se pueden encontrar mucha documentación técnica de soporte en la web; desde el aspecto económico, no es necesario la compra de licencias para su uso.

5.2.3. Metodología de desarrollo

En este apartado toca describir la representación gráfica del sistema desarrollado, representado por los casos de uso y sus diagramas correspondientes. En el desarrollo de la investigación se ha utilizado el Proceso Unificado Racional (RUP por si siglas en inglés) y el Lenguaje Unificado de Modelado (UML en inglés), conformando ésta la metodología orientada a objetos para el desarrollo de software.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Para llevar un orden, se va a llevar la secuencialidad de la metodología usada como sigue:

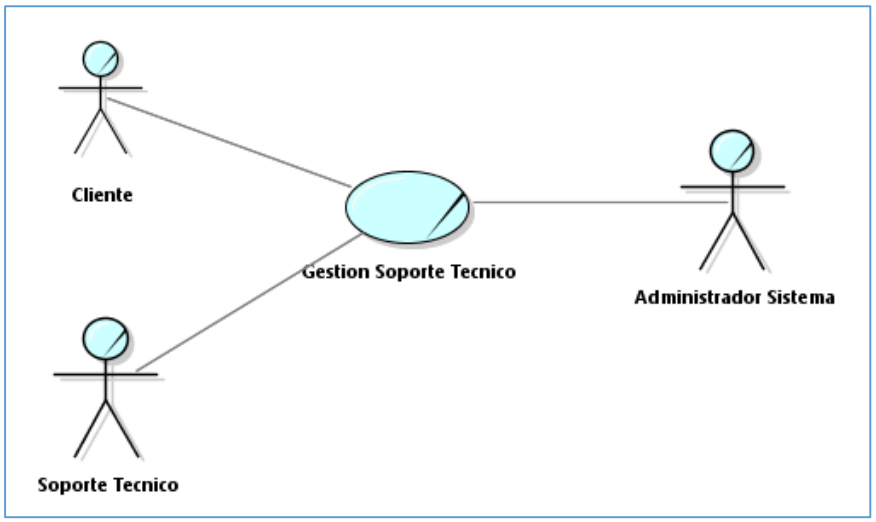
1. **Fase Inicial o de Concepción:**

Esta fase permitió definir el alcance del proyecto e identificando los casos de uso que describen las características y funcionalidades que son deseadas por cada clase importante del diseño.

a. **Modelo de Caso de Uso del Negocio.**

En este modelo de identifica los actores y casos de uso del negocio, permitiendo de manera general mostrar el proceso del negocio.

Figura 7 : Caso de uso del negocio



Fuente: Diseño propio.

Especificaciones del caso de uso del negocio

Cuadro 7 : Descripción especificación del caso de uso del negocio.

Gestión de Soporte técnico	
Definición	Proceso que consiste en gestionar los elementos que ayuden a realizar una consulta teniendo como soporte el procesamiento de lenguaje natural desde el enfoque de Soporte Técnico.
Metas	Atender eficientemente las consultas de los usuarios a través del procesamiento de lenguaje natural.
Actores	Soporte Técnico, Administrador de la aplicación, Cliente
Riesgos	Generar resultados con valores falsos, perjudicando a los usuarios.
Categoría	Caso de Uso principal.

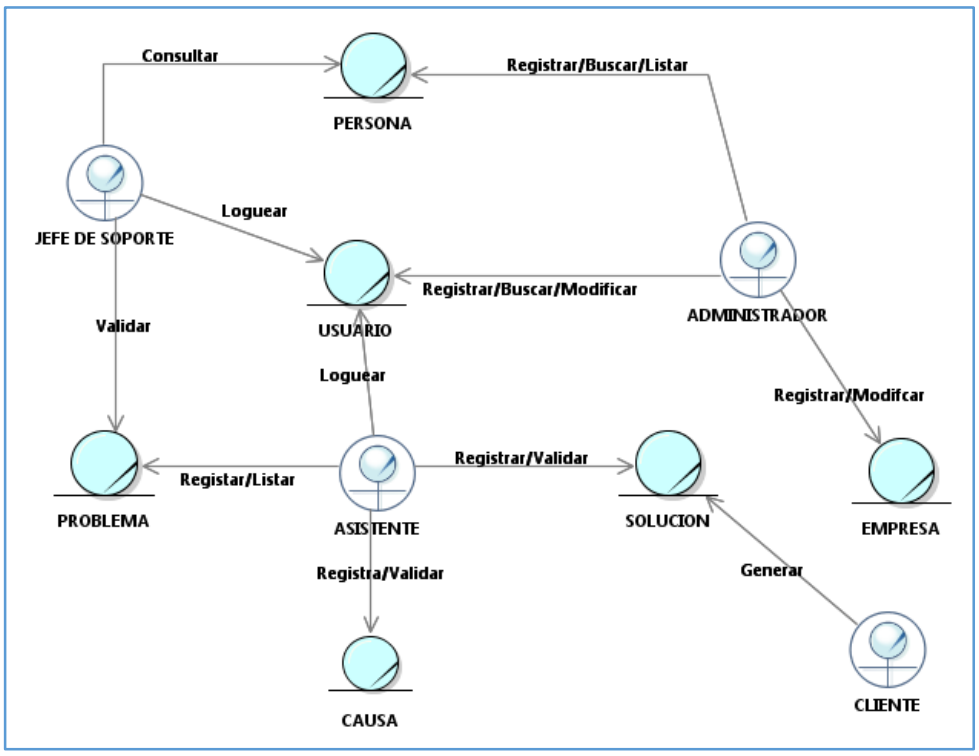
Fuente: Diseño propio.

b. Modelo de Objetos del Negocio.

Modelo que permitió identificar los roles y entidades en el negocio para el caso de uso identificado.

Gestión de Soporte técnico

Figura 8 : Diagrama Gestión de Soporte

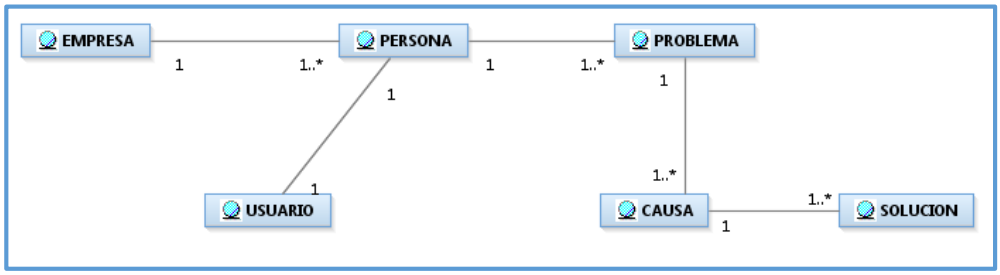


Fuente: Diseño propio.

c. Modelo del Dominio del Problema.

El modelo del dominio es un diagrama de clases que permitió captar los objetos que realizan actividades de entrada y salida en el sistema, siendo el punto de partida para el diseño de dicho sistema.

Figura 9 : Diagrama Dominio del Problema



Fuente: Diseño propio.

2. Fase de Elaboración:

En esta fase se obtuvieron los requerimientos funcionales y no funcionales que es una parte importante en el desarrollo de todo proyecto de software, así mismo se analizó el dominio del problema, estableciendo una arquitectura sólida y eliminando los elementos de mayor riesgo para el desarrollo exitoso del proyecto.

A. Requerimientos

Como requerimientos se lograron identificar y obtener todos los requerimientos funcionales aquellos que son cumplidas por el sistema y no funcionales que sirvieron para la parte operativa del sistema.

i. Requerimientos Funcionales

Cuadro 8 : Descripción Requerimientos funcionales.

Nº	Nombre del requerimiento	Prioridad
1	**Contar con:** Registrar y actualizar Empresa	**Alta**

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

	5.2.1. Registrar y actualizar Personal 5.2.2. Registrar y actualizar Usuarios 5.2.3. Registrar Problemas 5.2.4. Registrar Causas 5.2.5. Registrar Soluciones	
2	**Generar:** 5.2.6. Entrenamiento de las nuevas preguntas y soluciones del sistema. 5.2.7. El resultado de las consultas a través del procesamiento del lenguaje natural	**Alta**
3	**Consultar:** 5.2.8. Personal 5.2.9. Usuarios 5.2.10. Problemas	**Media**
4	**Realizar:** 5.2.11. Reporte de log de la búsqueda.	**Alta**

Fuente: Diseño propio.

ii. Requerimientos No funcionales

Implica la determinación de los atributos no funcionales asociadas a las facilidades, funcionalidades y de características generales del software.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Cuadro 9 : Descripción Requerimientos No funcionales.

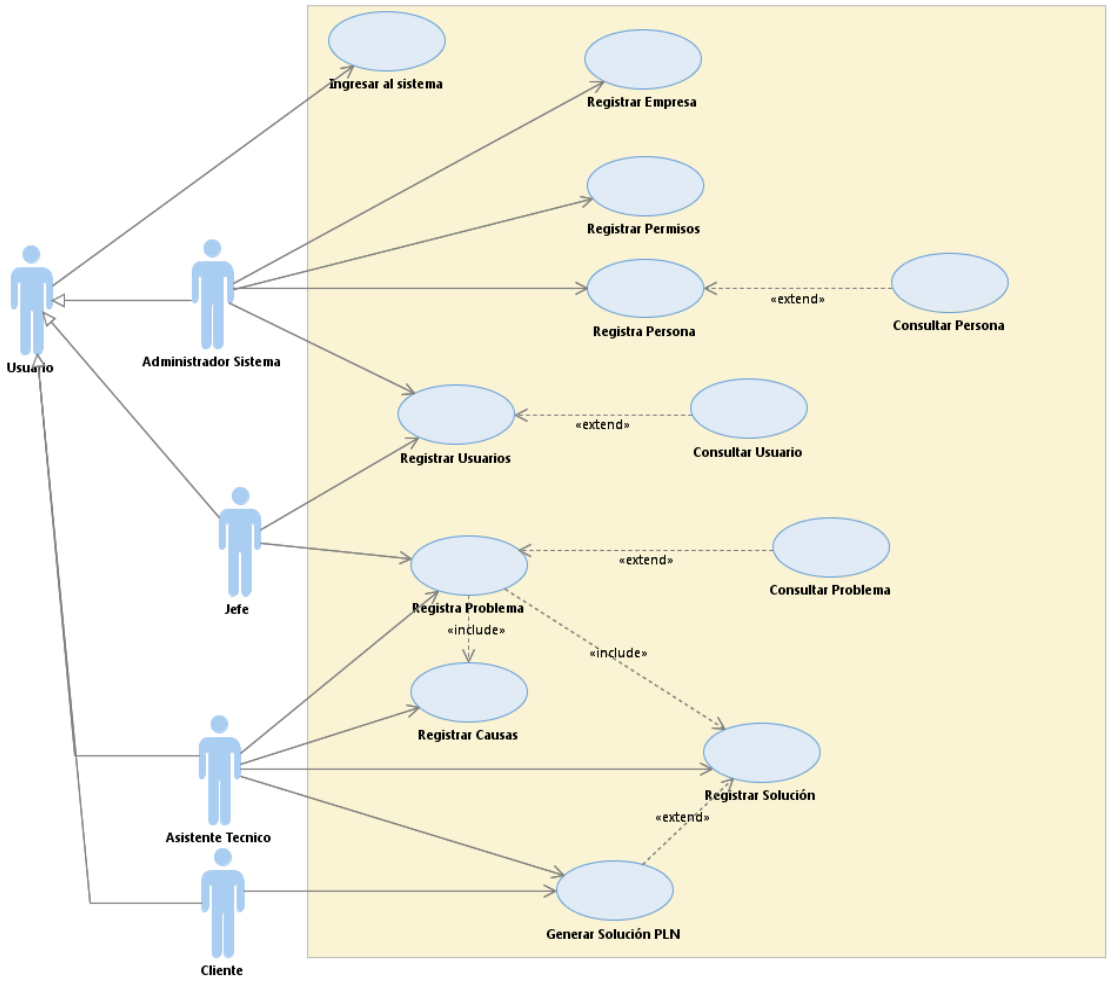
Nº	Nombre del requerimiento
1	Utilizar equipos de cómputo adecuados para el funcionamiento del sistema.
2	Utilizar el software de BD y de desarrollo adecuado para su implementación.
3	Utilizar el sistema operativo adecuado para la implementación.
4	Capacitar al personal en el uso de la aplicación.
5	Contar con el hardware adecuado para la impresión, si así lo requiera.
6	Debe poderse instalar toda la aplicación en una equipo compartido.

Fuente: Diseño propio.

iii. Modelo de casos de uso de Requerimiento (MCUR)

Permitió capturar los requerimientos para especificar las interacciones entre los actores y el sistema:

Figura 10 : Diagrama Casos de uso de requerimiento.



Fuente: Diseño propio.

Especificación Caso de Uso: Registrar Empresa

Cuadro 10 : Descripción Caso de Uso: Registrar Empresa.

Especificación Caso de Uso: Registrar Empresa	
Descripción	Datos principales para la empresa.
Pre-Condición	• El Administrador debe haber ingresado al sistema.
	Flujo Básico

Flujo de Eventos	**1.** El Administrador solicita al sistema comenzar con el proceso de registro de la empresa. **2.** El Sistema solicita los siguientes datos: Nombre, dirección, teléfono, correo. **3.** El sistema almacena los datos proporcionados.
	Flujo Alternativo
	4. En el punto Nº2 en el caso que ya se encuentre registrado pueden modificar los datos para que estos sean actualizados en el sistema.
Post-Condición	Valida empresa en la entidad.

Fuente: Diseño propio.

Especificación Caso de Uso: Registrar Usuario

Cuadro 11 : Descripción Caso de Uso: Registrar Usuario.

Especificación Caso de Uso: Registrar Usuario	
Descripción	Se realiza el registro de usuarios, para darle el acceso al personal encargado al sistema.
Pre-Condición	• El administrador debe haber ingresado al sistema. • Los datos de la persona.
Flujo de Eventos	**Flujo Básico**

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

	1. El administrador solicita al sistema comenzar con el proceso de registro de Usuario. **2.** El administrador ingresar en el área a la que será asignado el usuario. **3.** El administrador deberá elegir la opción de Nuevo usuario. **4.** El Sistema solicita los siguientes datos: Persona, usuario, contraseña, estado. **5.** El sistema almacena los datos proporcionados.
	Flujo Alternativo
	6. En el punto Nº2 el administrador puede previamente realizar la búsqueda si la persona ya cuenta con usuario. **7.** En el punto Nº3 el Administrador antes de elegir la opción Nuevo, podrá acceder a la opción editar si en caso lo requiera.
Post-Condición	Valida a usuarios en la entidad. Autogenera el código de usuario.

Fuente: Diseño propio.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Especificación Caso de Uso: Registrar Persona

Cuadro 12 : Descripción Caso de Uso: Registrar Persona.

Especificación Caso de Uso: Registrar Persona	
Descripción	Se realiza el registro de la persona para que se le asigne un usuario.
Pre-Condición	El Administrador debe haber ingresado al sistema.
Flujo de Eventos	**Flujo Básico** **1.** El administrador solicita al sistema comenzar con el proceso de registro de persona. **2.** El administrador deberá elegir la opción de Nuevo. **3.** El Sistema solicita los siguientes datos: Nombres, apellidos, sexo, dirección, teléfono, celular, email, ubigeo, número del documento, estado civil. **4.** El sistema almacena los datos proporcionados. **Flujo Alternativo** **5.** En el punto Nº 2 el administrador puede previamente realizar la búsqueda de la persona, si desconoce si se encuentra o no en el registro.
Post-Condición	Autogenera el código de la persona.

Fuente: Diseño propio.

Especificación Caso de Uso: Registrar Problema, Causa y Solución

Cuadro 13 : Descripción Caso de Uso: Registrar Problema, Causa y Solución.

Especificación Caso de Uso: Registrar Problema, Causa y Solución	
Descripción	Se realiza el registro de los problemas de soporte tecnológico.
Pre-Condición	El Jefe o el asistente técnico, debe haber ingresado al sistema.
Flujo de Eventos	**Flujo Básico**
	1. El Jefe o el asistente técnico solicita al sistema comenzar con el proceso de registro de persona. **2.** El Jefe o el asistente técnico deberá elegir la opción de Nuevo. **3.** El Sistema solicita los siguientes datos: Problema, causas, soluciones. **4.** El sistema almacena los datos proporcionados.
	Flujo Alternativo
	5. En el punto Nº2 el jefe o el asistente técnico, puede previamente realizar la búsqueda del problema, si desconoce si se encuentra o no en el registro.
Post-Condición	Autogenera el código de la problema, causas y soluciones.

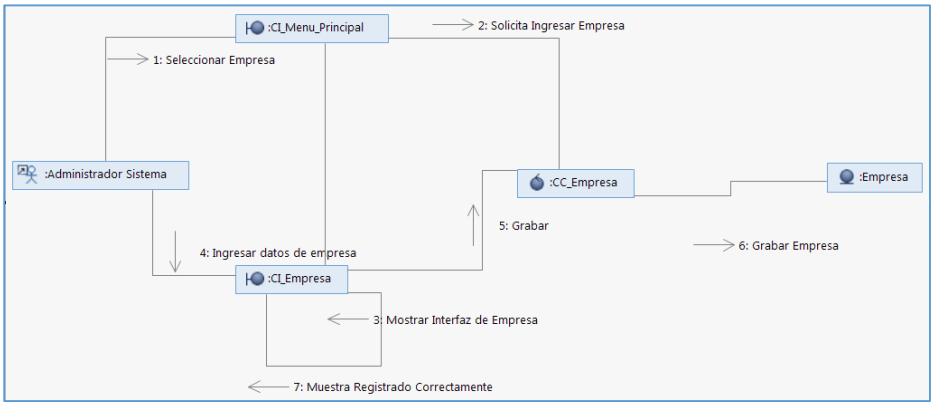
Fuente: Diseño propio.

B. Diagrama de colaboración

Los diagramas de colaboración o llamados también diagramas de comunicación muestran las interacciones y los enlaces entre un conjunto de objetos que colaboran entre sí, centrándose en el espacio mostrando el contexto de la operación y ciclos de la ejecución.

Diagrama de Colaboración: Registrar Empresa

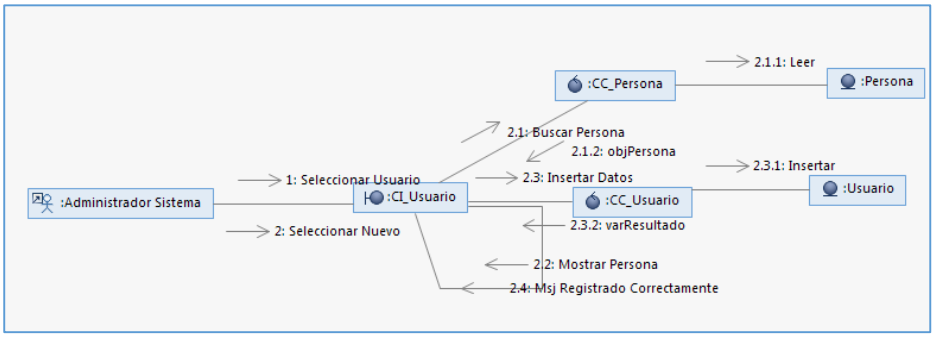
Figura 11 : Diagrama de Colaboración: Registrar Empresa.



Fuente: Diseño propio.

Diagrama de Colaboración: Registrar Usuario

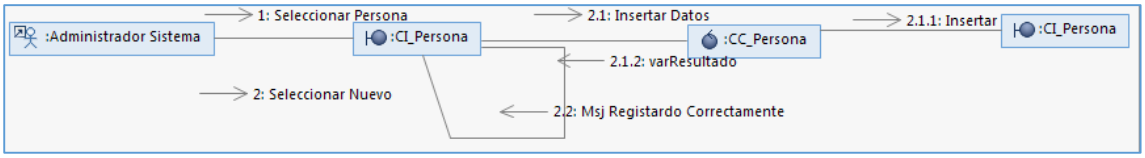
Figura 12 : Diagrama de Colaboración: Registrar Usuario.



Fuente: Diseño propio.

Diagrama de Colaboración: Registrar Persona

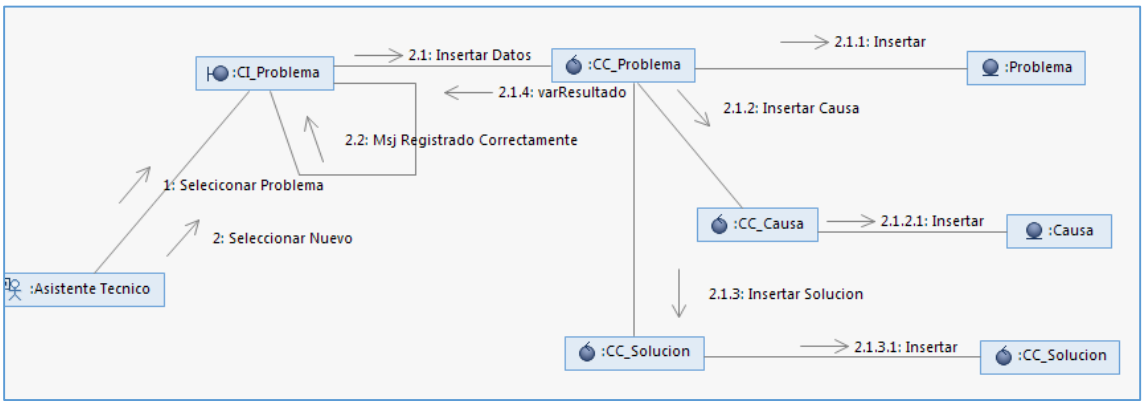
Figura 13 : Diagrama de Colaboración: Registrar Persona.



Fuente: Diseño propio.

Diagrama de Colaboración: Registrar Problema, Causa y Solución

Figura 14 : Diagrama de Colaboración: Registrar Problema, Causa y Solución.



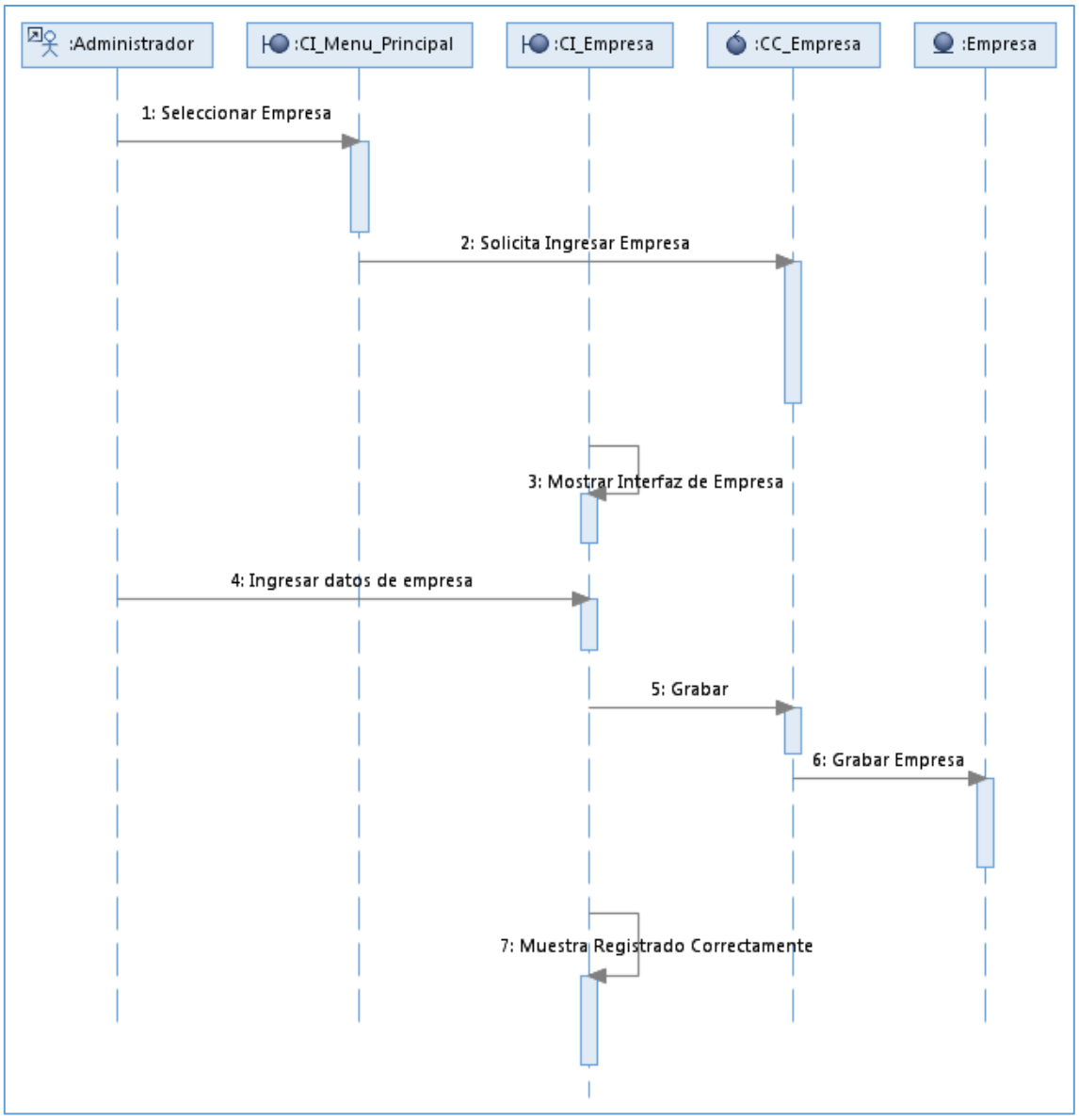
Fuente: Diseño propio.

C. Diagrama de Secuencia

Con estos diagramas se han identificado la interacción que tienen los objetos entre ellos y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino.

Diagrama de Secuencia: Registrar Empresa

Figura 15 : Diagrama de Colaboración: Registrar Empresa.

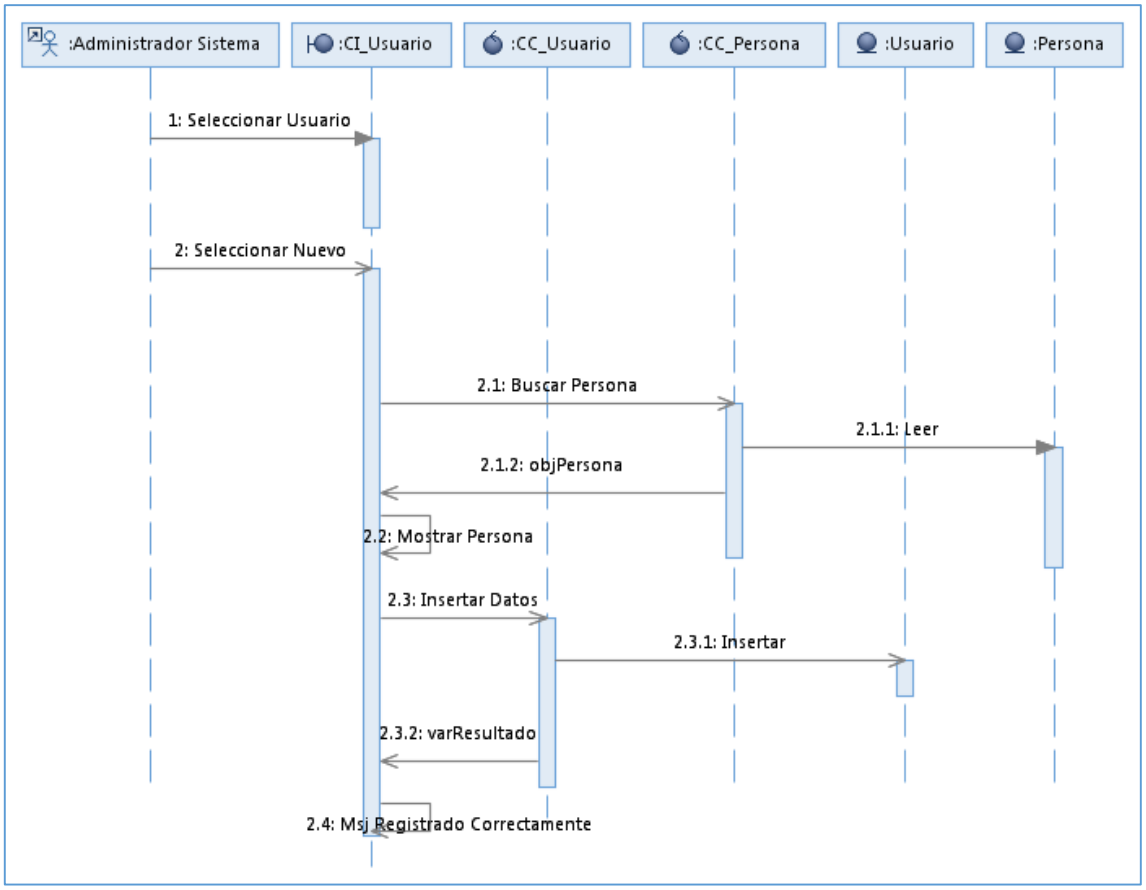


Fuente: Diseño propio.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Diagrama de Secuencia: Registrar Usuario

Figura 16 : Diagrama de Colaboración: Registrar Usuario.

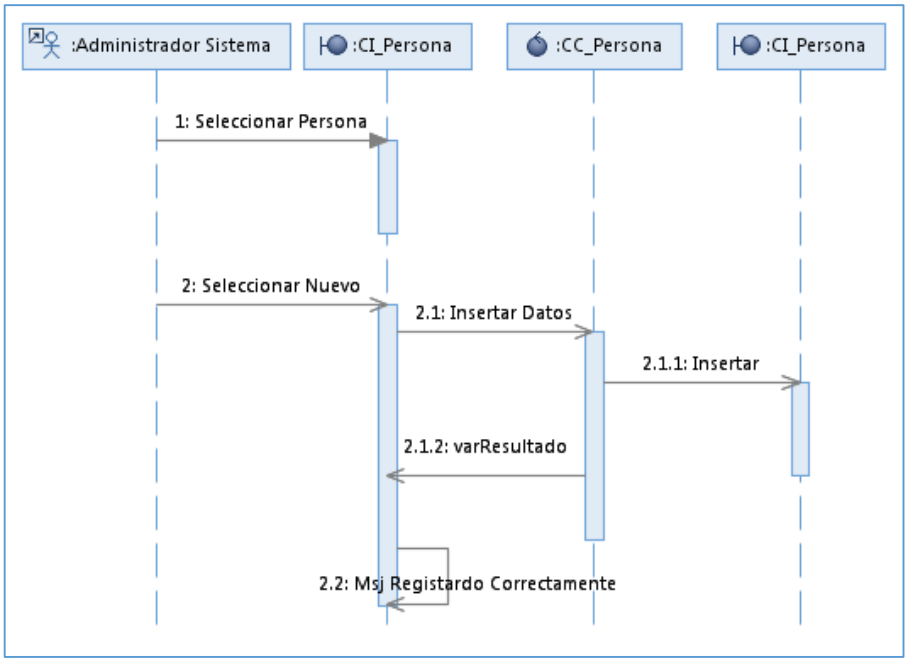


Fuente: Diseño propio.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Diagrama de Secuencia: Registrar Persona

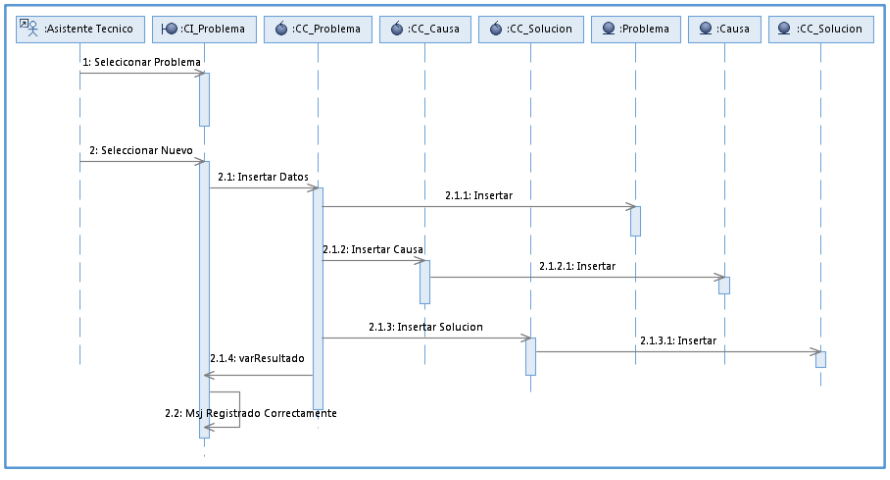
Figura 17 : Diagrama de Colaboración: Registrar Persona.



Fuente: Diseño propio.

Diagrama de Secuencia: Registrar Problema, Causa y Solución

Figura 18 : Diagrama de Colaboración: Registrar Problema, Causa y Solución.



Fuente: Diseño propio.

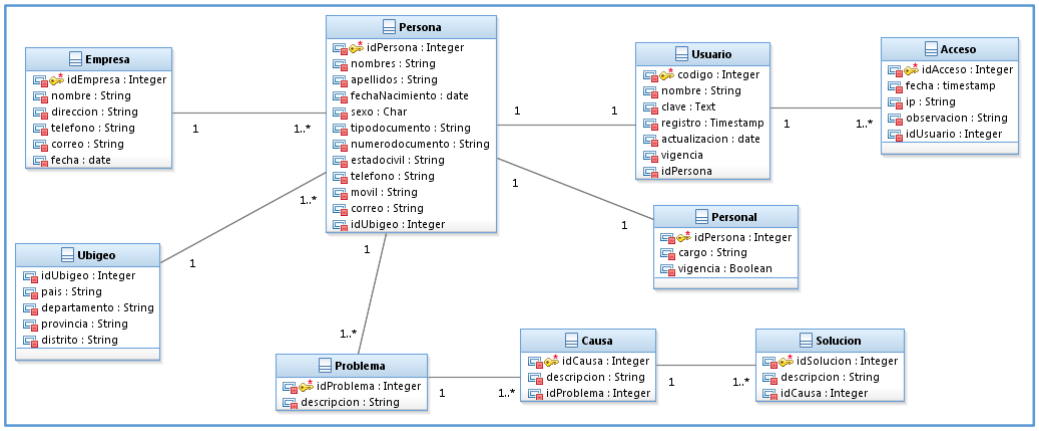
Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3. Fase de Construcción:

A. Diagrama de Clases

Se identifican las clases que estructura el sistema desarrollado, mostrando sus atributos y sus tipos de relaciones.

Figura 19 : Diagrama de Clases.



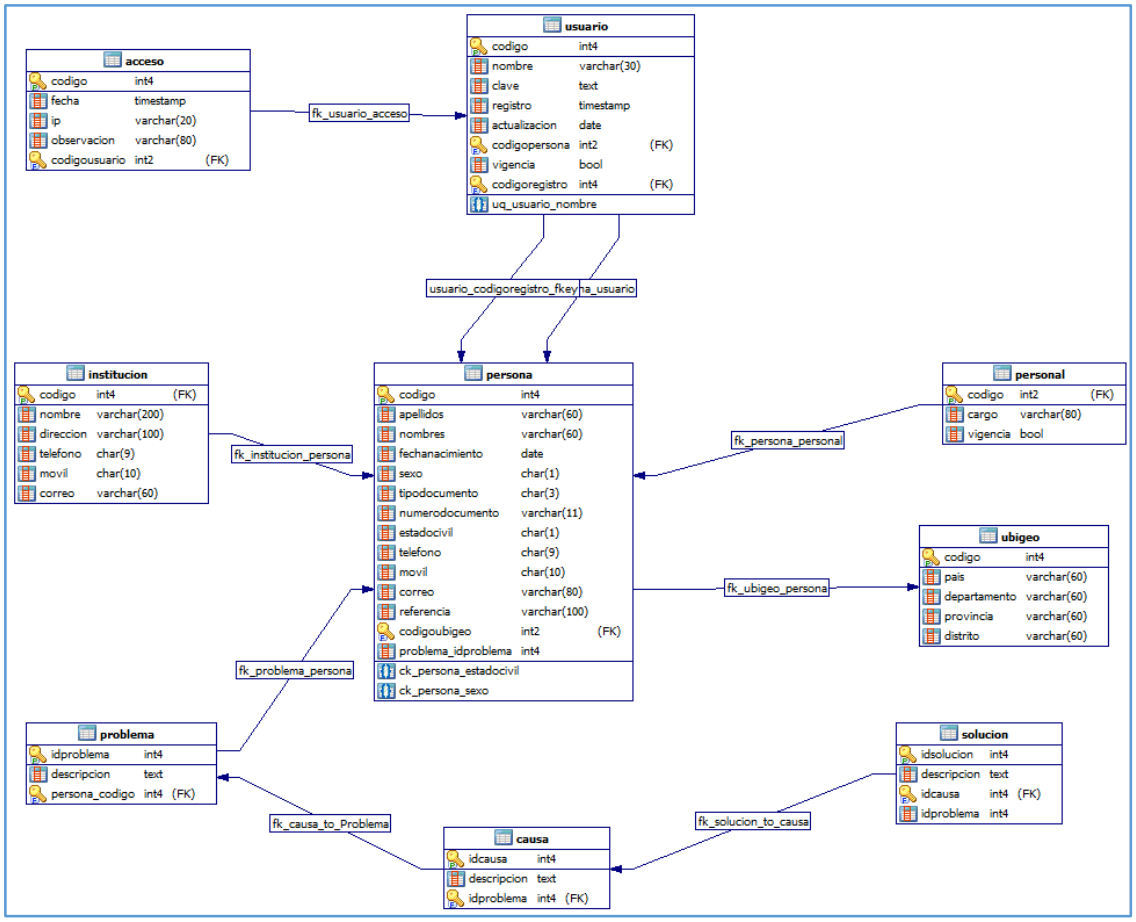
Fuente: Diseño propio.

B. Diagrama de Base de Datos

En el siguiente diagrama se presenta como ha sido diseñada la base de datos considerando el diagrama de clases.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 20 : Diagrama de Base de datos.



Fuente: Diseño propio.

C. Diagrama de Componentes

Con el diagrama de componentes se va a mostrar de manera lógica la arquitectura de software.

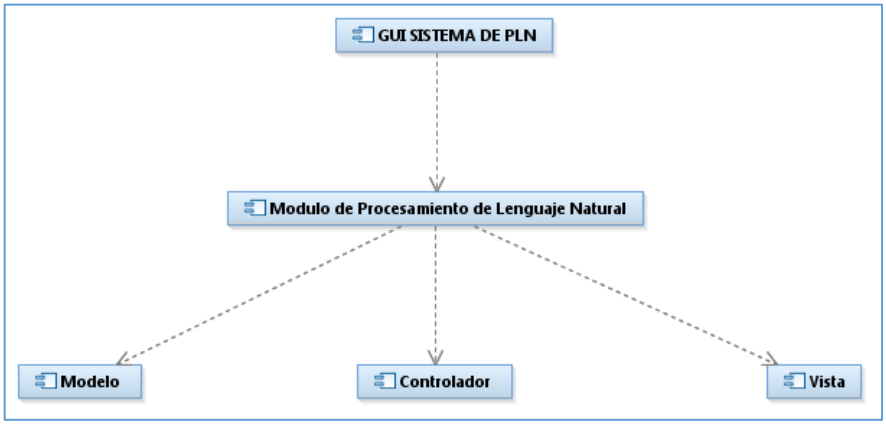
Componentes del Sistema

En la siguiente Imagen se puede ver que el sistema solo cuenta con un módulo, pero así mismo ha sido desarrollado con

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

la arquitectura de 3 capas, y con la arquitectura del MVC (Modelo, Vista, controlador).

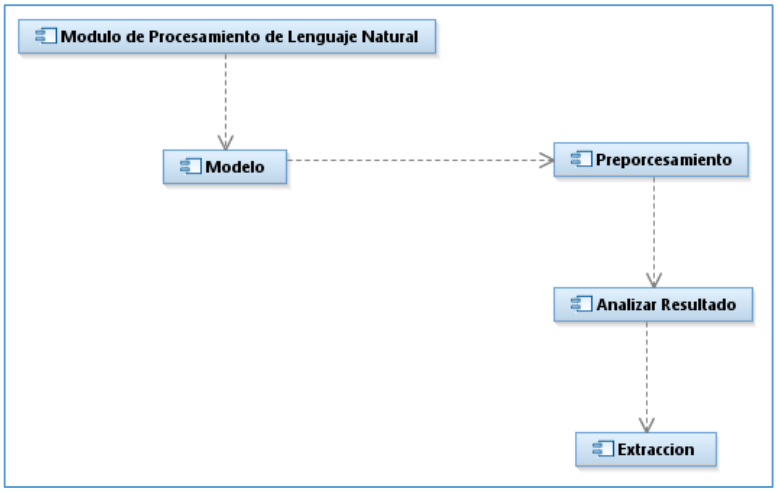
Figura 21 : Diagrama de Componentes del Sistema.



Fuente: Diseño propio.

Componentes del Algoritmo PLN.

Figura 22 : Diagrama del Algoritmo PLN.



Fuente: Diseño propio.

D. Diagrama de Despliegue

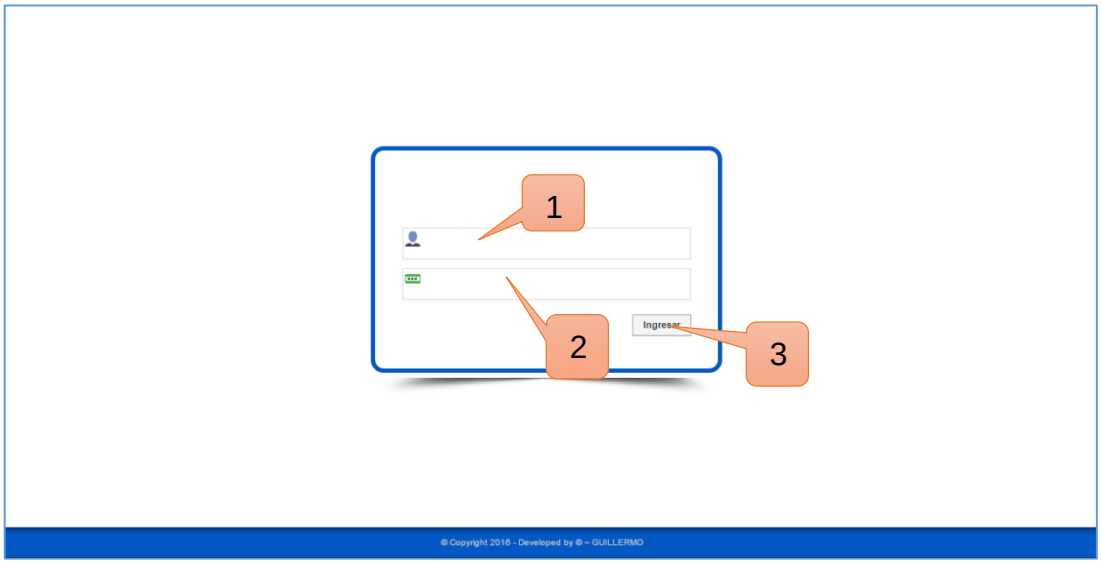
El objetivo principal de este diagrama es mostrar la disposición de las particiones físicas del sistema de información y la asignación de los componentes software a estas particiones. Es decir, las relaciones físicas entre los componentes software y hardware en el sistema desarrollado.

E. Interfaces del Sistema

A continuación, se muestran las pantallas o interface del sistema, asimismo se describirá su uso, para que dé una idea más clara de lo que se verá al momento de la ejecución del mismo.

Inicio de Sesión del Sistema

Figura 23 : Pantalla de inicio de sesión.



Fuente: Diseño propio.

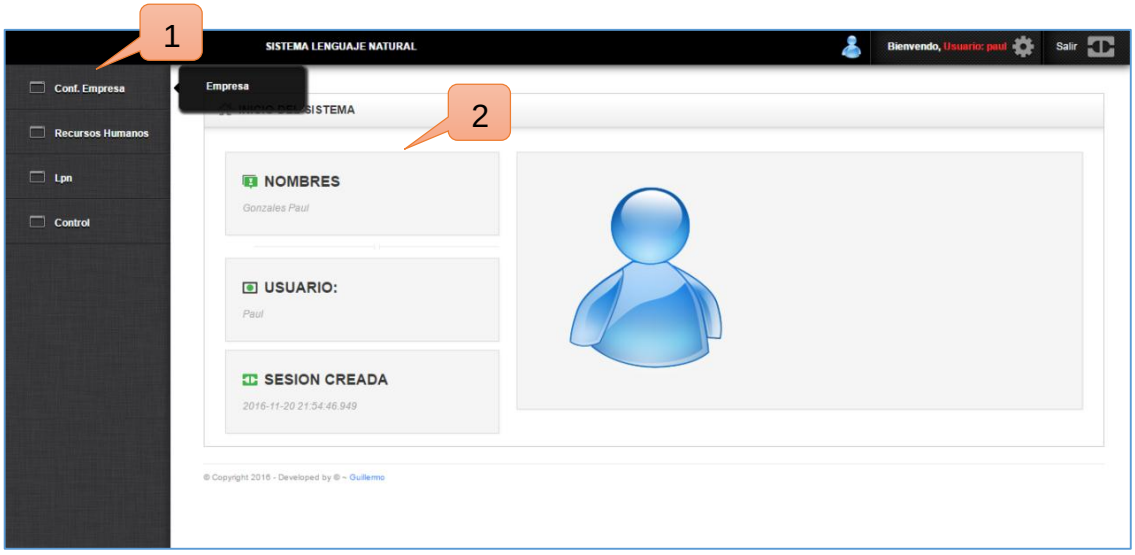
1. Ingresar el usuario de acceso generado por el administrador de la aplicación.
2. Se debe ingresar la clave correspondiente.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3. Hacer clic en el botón ingresar. Si no existiese el usuario o la clave de acceso fuese incorrecta, la aplicación mostrará un mensaje.

Página principal

Figura 24 : Pantalla de Pantalla principal.



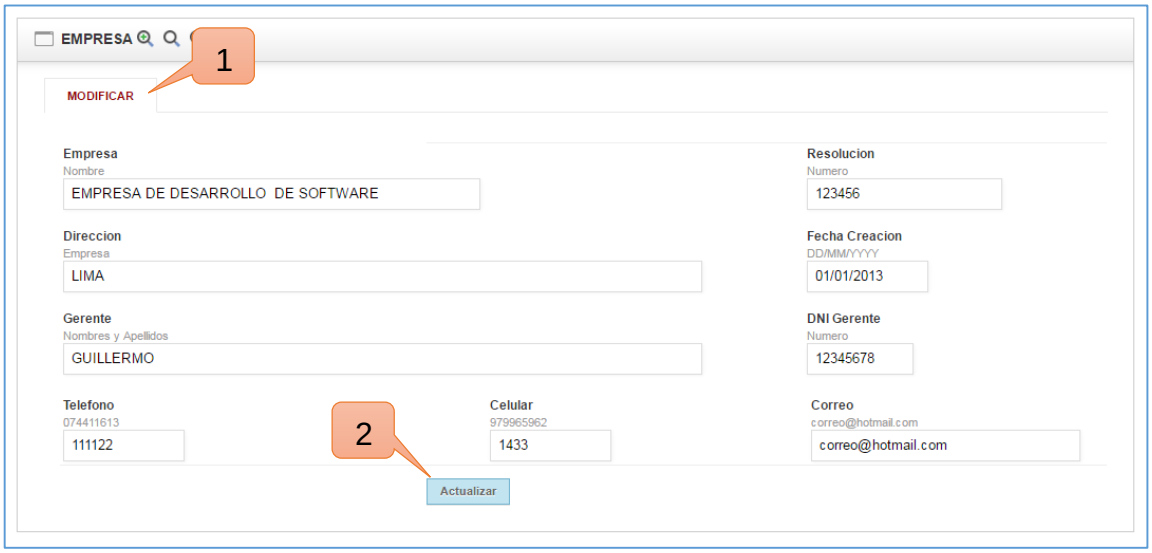
Fuente: Diseño propio.

1. En esta parte de la pantalla muestra el menú contextual de las opciones con las que cuenta el sistema.
2. Como se visualiza en la imagen en esta zona de la pantalla se va a poder visualizar los datos del usuario que ingresó a la aplicación.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Interface Mantenimiento Empresa

Figura 25 : Pantalla de Mantenimiento Empresa.

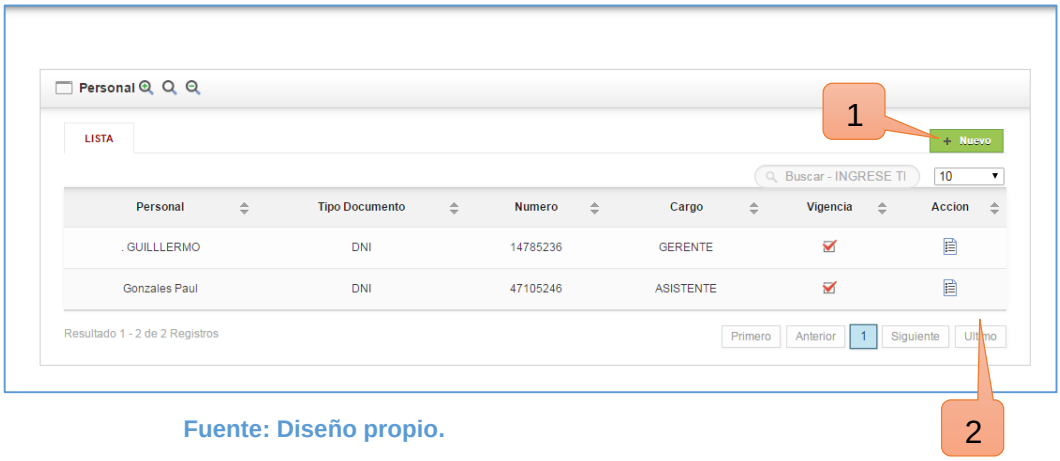


Fuente: Diseño propio.

1. Se muestra la pantalla de mantenimiento, cabe indicar que de acuerdo a la acción que se haga (Nuevo o Modificar), ésta mostrará los datos que se visualizan para el registro o cambio que amerite.
2. Una vez que se tengan los cambios realizados se deberá dar clic al botón “Actualizar” para grabar los datos en la base de datos.

Interface Mantenimiento Personal

Figura 26 : Pantalla de Mantenimiento Personal - Vista.



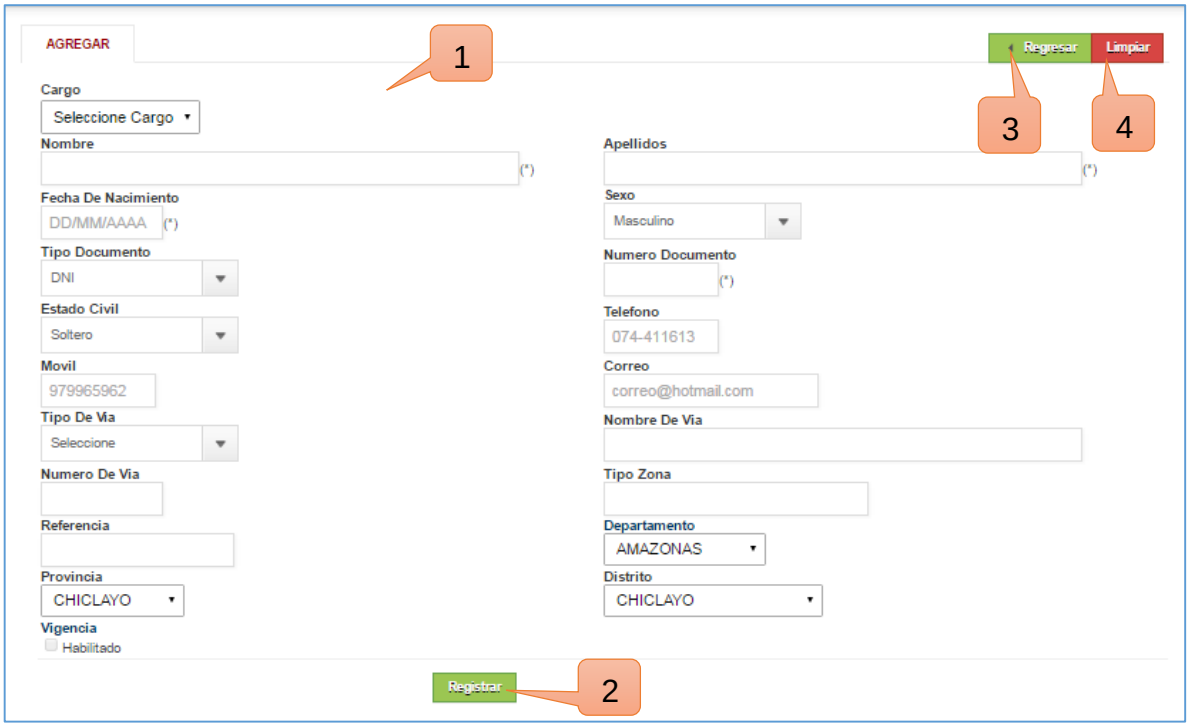
Fuente: Diseño propio.

Esta opción mostrará una lista inicial del personal registrado en la aplicación.

1. Para poder agregar información se deberá dar clic en el botón “Nuevo”.
2. Para modificar datos del personal, se debe ubicar el registro que se requiere modificar y se hace clic en el botón indicado en la figura.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 27 : Pantalla de Mantenimiento Personal - Agregar.



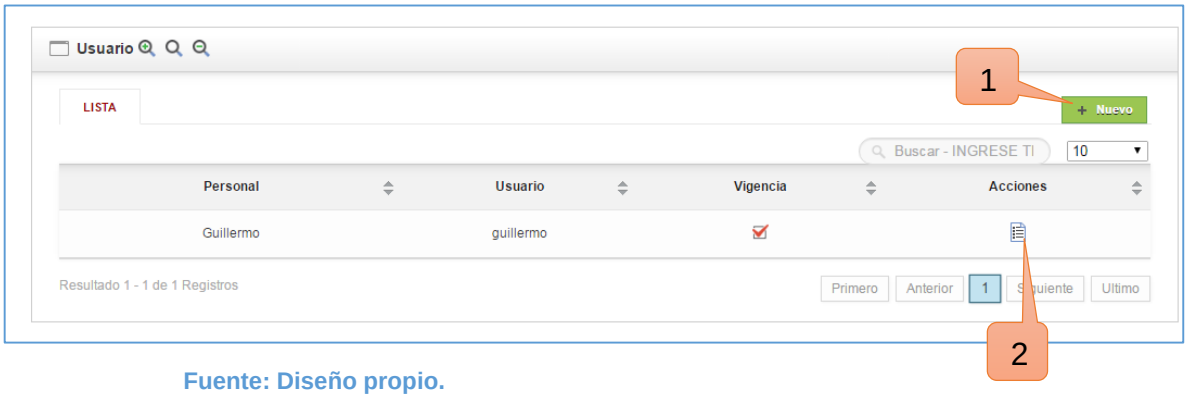
Fuente: Diseño propio.

1. Para agregar o modificar información mostrará esta pantalla con datos con los que cuenta el personal. Se deben registrar los datos necesarios. Considerar que los campos que están identificados con un asterisco (“*”) son obligatorios.
2. Cuando se tengan los datos registrados solo se debe hacer clic en el botón registrar.
3. Se hará clic en este botón (“Regresar”), si se necesita salir de la ventana sin grabar.
4. Para limpiar los datos de la ventana se dará clic en el botón “Limpiar”.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Interface Mantenimiento Usuario

Figura 28 : Pantalla de Mantenimiento Usuario - Vista.



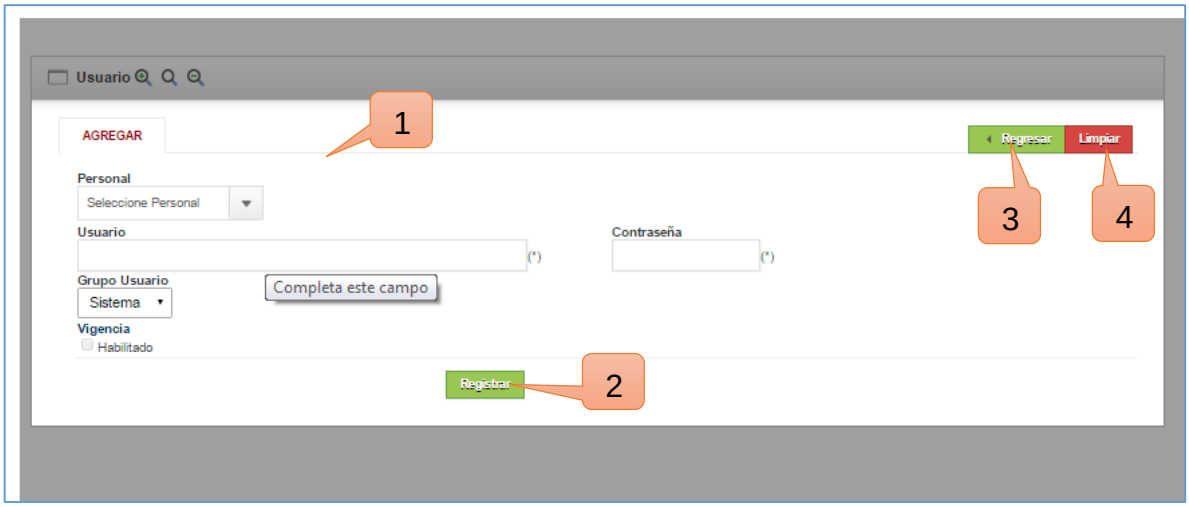
Fuente: Diseño propio.

Esta opción mostrará una lista inicial de los usuarios registrados en la aplicación.

1. Para poder agregar información se debe dar clic en el botón "Nuevo".
2. Para modificar datos de los usuarios, se debe ubicar el registro que se requerirá modificar y dar clic en el botón indicado en la figura.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Figura 29 : Pantalla de Mantenimiento Usuario - Agregar.



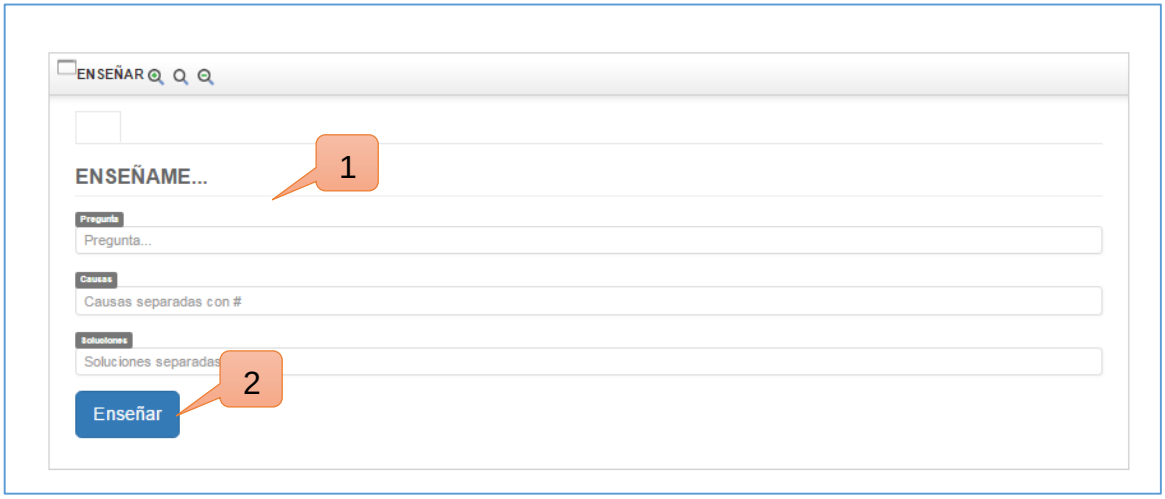
Fuente: Diseño propio.

1. Para agregar o modificar información se mostrará esta pantalla con datos con los que cuenta el usuario. Se debe registrar los datos necesarios. Considerar que los campos que están identificados con un asterisco (“*”) son obligatorios.
2. Cuando se tienen los datos registrados solo deben hacer clic en el botón registrar.
3. Se debe clic en este botón (“Regresar”) si se necesita salir de la ventana sin grabar.
4. Para limpiar los datos de la ventana se debe hacer clic en el botón “Limpiar”.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Interface Mantenimiento Problema, Causa y Soluciones.

Figura 30 : Pantalla de Mantenimiento Problema, Causa y Soluciones.



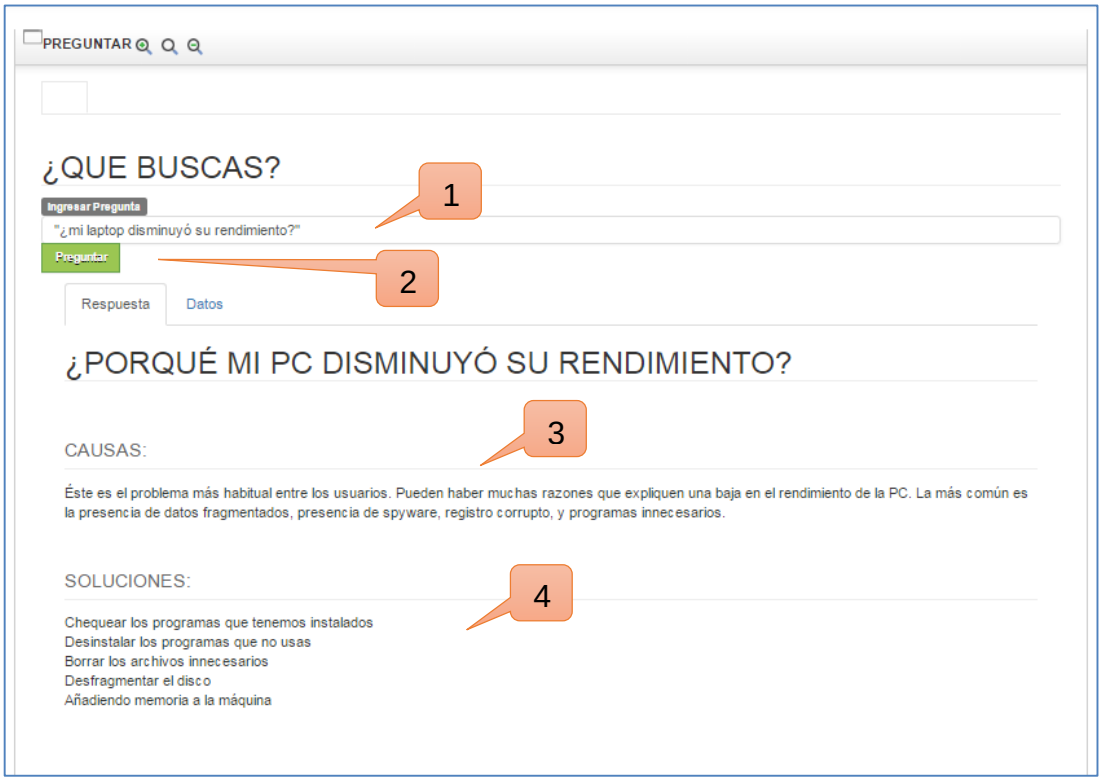
Fuente: Diseño propio.

1. Para agregar información de problemas causas y soluciones se mostrará esta pantalla. Se deberá registrar los datos necesarios, registrándolo conforme lo pide la pantalla.
2. Una vez que se tienen los datos registrados solo se debe hacer clic en el botón “enseñar” y la información se registrará en la base de datos de conocimiento.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Consultar solución por PLN

Figura 31 : Pantalla de Consulta de solución por PLN.



Fuente: Diseño propio.

1. Como primer paso, se deberá ingresar la consulta en lenguaje natural.
2. Para hacer la búsqueda se hará clic en el botón “Preguntar”, la misma que será procesada por el motor de búsqueda.
3. En esta zona se mostrará los datos de las causas relacionadas a la consulta.
4. En esta zona se mostrará los datos de la solución relacionadas a la consulta.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

5.3. Detalle técnico del método de procesamiento de consultas.

En este apartado, se dará una descripción de las técnicas usadas en el motor de procesamiento de consultas, como son los algoritmos **LCS** y **Levenshtein,** los cuales trabajan en conjunto como parte de este método; asimismo se describirá la forma de cómo se integraron para la generación del motor de búsqueda y que se encuentra en la Lógica de negocio. Seguido se dan los detalles:

5.3.1. Algoritmo LCS (Longest Common subsequence).

El algoritmo LCS o Subsecuencia común más larga, se puede describir el problema de la siguiente manera (Carnegie Mellon University - Computer Science departament, 2015). Se dan dos cadenas: Cadena S de longitud “n”, y cadena T de longitud “m”. El objetivo es producir su subsecuencia común más larga de caracteres que aparecen de izquierda a derecha (pero no necesariamente en un bloque contiguo) en ambas cadenas.

Por ejemplo, considera:

S = ABAZDC

T = BACBAD

En este caso, el LCS tiene longitud 4 y es la cadena ABAD. Otra forma de verlo es que se está buscando una coincidencia de 1-1 entre algunas de las letras en S y algunas de las letras en T, de modo que ninguno de los bordes en la coincidencia se cruzan entre sí.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Por ejemplo, este tipo de problema aparece todo el tiempo en genómica: Dados dos fragmentos de DNA, el LCS brinda información sobre lo que tienen en común y la mejor forma de alinearlos.

Se da la solución ahora al problema de LCS usando la Programación Dinámica. Como sub problemas se verá el LCS de un pre x de S y un pre x de T, corriendo sobre todos los pares de pre xes. Para simplificar, existe la preocupación primero por conocer la longitud del LCS y luego se puede modificar el algoritmo para producir la secuencia real.

Entonces, aquí está la pregunta: se dice que LCS [i, j] es la longitud del LCS de S [1...i] con T [1...j]. ¿Cómo se puede resolver para LCS [i, j] en términos de los LCS de los problemas más pequeños?

Caso 1: ¿Que si S[i] 6 ≠ T[j]? Entonces, la sub secuencia deseada tiene que ignorar uno de S [i] o T [j] así que se tiene:

$$LCS [i; j] = \max (LCS [i - 1; j]; LCS [i; j - 1])$$

Caso 2: ¿y si S [i] = T [j]? Entonces, el LCS de S [1…i] y T [1…j] podría coincidir con ellos. Por ejemplo, si le diera una subsecuencia común que coincidiera con S [i] a una ubicación anterior en T, por ejemplo, siempre podría hacerla coincidir con T [j]. Entonces, en este caso se tiene:

$$LCS [i; j] = 1 + LCS [i - 1; j -1]$$

Entonces, se podrá hacer dos bucles (sobre los valores de i y j), cargando en el LCS usando estas reglas. Esto es lo que parece ilustrado

para el ejemplo anterior, con S a lo largo de la columna más a la izquierda y T a lo largo de la fila superior.

Cuadro 14 : Descripción proceso LCS.

	B	A	C	B	A	D
A	0	1	1	1	1	1
B	1	1	1	2	2	2
A	1	2	2	2	3	3
Z	1	2	2	2	3	3
D	1	2	2	2	3	4
C	1	2	3	3	3	4

Fuente: LCS (Carnegie Mellon University - Computer Science departament, 2015)**.**

Se acaba de sacar esta matriz fila por fila, haciendo una cantidad constante de trabajo por entrada, por lo que toma O (mn) el tiempo total. La respuesta final (la longitud del LCS de S y T) está en la esquina inferior derecha.

¿Cómo se puede encontrar ahora la secuencia? Para encontrar la secuencia, simplemente se camina hacia atrás a través de la matriz comenzando por la esquina inferior derecha. Si la celda directamente arriba o directamente a la derecha contiene un valor igual al valor en la celda actual, entonces debe moverse a esa celda (si ambas son, entonces elija una). Si ambas celdas tienen valores estrictamente menores que el valor de la celda actual, mueva diagonalmente hacia la izquierda (esto corresponde a la aplicación del Caso 2), y genere el carácter asociado. Esto generará los caracteres en el LCS en orden inverso. Por ejemplo, ejecutándose en la matriz de arriba, esto produce DABA.

5.3.2. Algoritmo Levenshtein.

El algoritmo de Levenshtein permite encontrar la distancia mínima de edición (Rodríguez, 2012). Este algoritmo considera que las operaciones de borrado e inserción tienen un costo de 1 mientras que las de sustitución tienen un costo de 2 ya que la considera como compuesta por una operación de borrado y una de inserción. Así se tiene:

Inserción → 1

Borrado → 1

Sustitución → 2

Inicialización del Array

D (i, 0) = i

D (0, j) = j

Relación recurrente para el cómputo de la distancia

Para cada i de 1 a M

Para cada j de 1 a N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 0 \text{ si } X(i) = Y(j) \\ 2 \text{ si } X(i) \neq Y(j) \end{cases} \end{cases}$$

Finalización

La distancia es D(N,M).

Explicación

La distancia se calcula recursivamente, donde la distancia entre la palabra A y la palabra B es la suma de la operación final (inserción, borrado o sustitución) más la distancia entre la palabra A-1 letra y la palabra B, o bien entre A y B-1 letra o bien entre la palabra A-1 y B-1 letra. Y así sucesivamente hasta llegar a una sola letra en donde las distancias son las obvias que se setean en la inicialización.

Este algoritmo se puede realizar también de forma iterativa, utilizando una tabla y muchas veces es útil hacerlo así porque interesa seguir la pista de las operaciones realizadas.

Pseudocódigo

Figura 32 : Ejemplo de Pseudocódigo – Algoritmo Levenshtein

```
Inicialización
matrix = new matrix [word1.length + 1][word2.length + 1] //Se define una matriz de enteros
 for(i=0;  i<=word1.length; i++)     matrix [i][0]=i; //Se inicializa las distancias de la fila 0,
 for(j=0; j<=word2.length; j++ )    matrix [0][j]=j; // Se inicializa las distancias de la columna 0

Computo
 for(i=1; i<=word1.length; i++)
          for(j=1;j<=word2.length; j++)

           if (word1[i-1] == word2[j-1]) factor = 0; //son iguales no hay costo de operación
  else  factor = 2;  //sustitucion

           matrix[i][j] = MIN(matrix[i-1][j]+1, matrix[i][j-1]+1, matrix[i-1][j-1]+factor);

Finalización
         return matrix[word1.length][word2.length];
```

Fuente: (Rodríguez, 2012)

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Ejemplo

Figura 33: Ejemplo de Pseudocódigo – Algoritmo Levenshtein – Sec. 1

N	9									
O	8									
I	7									
C	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	J	E	C	U	C	I	Ó	N

$$D(i,j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 0 \text{ si } X(i) = Y(j) \\ 2 \text{ si } X(i) \mathrel{!}= Y(j) \end{cases} \end{cases}$$

Fuente: (Rodríguez, 2012)

Figura 34: Ejemplo de Pseudocódigo – Algoritmo Levenshtein – Sec. 2

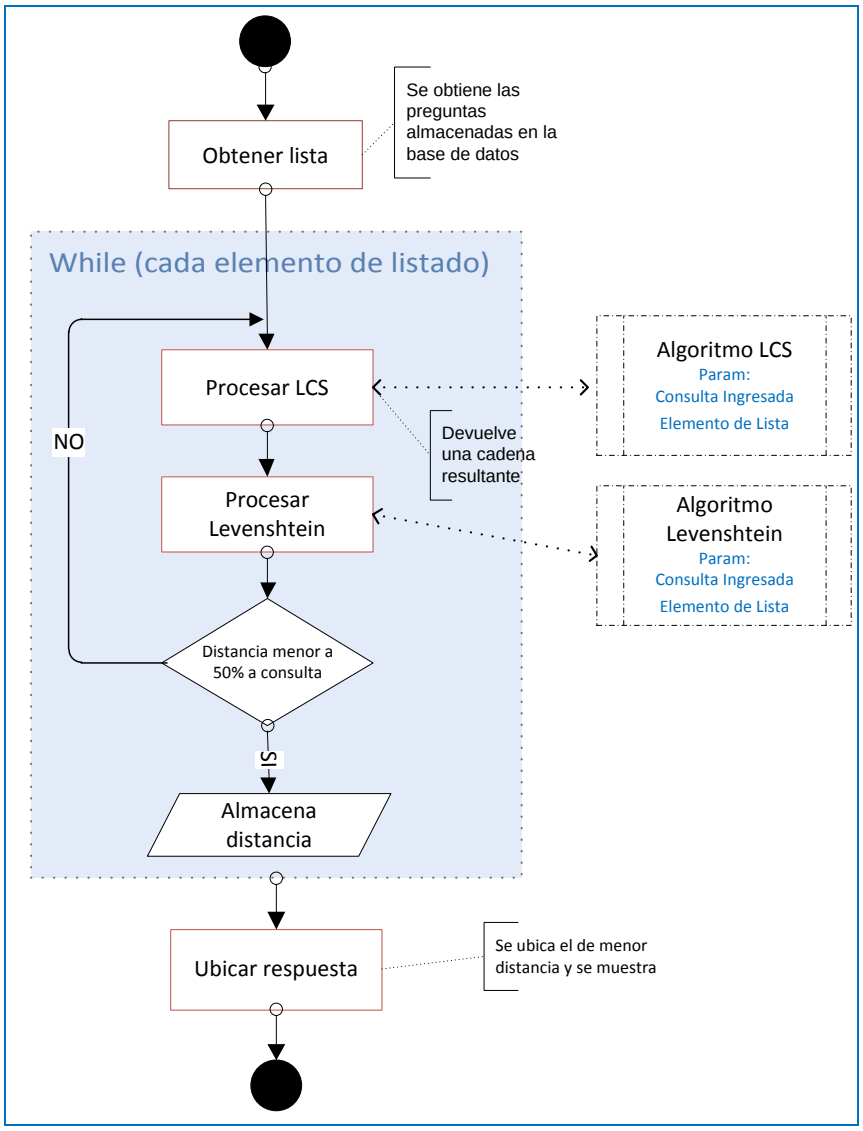
N	9	8	9	10	11	12	11	10	9	8
O	8	7	8	9	10	11	10	9	8	9
I	7	6	7	8	9	10	9	8	9	10
C	6	5	6	7	8	9	8	9	10	11
N	5	4	5	6	7	8	9	10	11	10
E	4	3	4	5	6	7	8	9	10	9
T	3	4	5	6	7	8	7	8	9	8
N	2	3	4	5	6	7	8	7	8	7
I	1	2	3	4	5	6	7	6	7	8
#	0	1	2	3	4	5	6	7	8	9
	#	E	J	E	C	U	C	I	Ó	N

Fuente: (Rodríguez, 2012)

5.3.3. Integración de los algoritmos LCS y Levenshtein.

En este punto se comparte y describe de manera resumida de cómo se logra enlazar los dos algoritmos para complementarlos en el motor de búsqueda.

Figura 35 : Gráfico de Integración LCS - Levenshtein.



Fuente: Diseño propio.

Explicación del Analizador

Para poder integrar los dos algoritmos se ha generado un método llamado Analizador donde se pueden identificar cuatro procesos importantes:

Primero: Se obtiene las preguntas almacenadas en la base de datos

Aquí lo que se realiza es listar todas las preguntas almacenadas de la Base de datos y se almacén en un array. Luego se revisa que, si la pregunta realizada coincide con alguna de las preguntas de la base de datos, si esta condición se cumple, devuelve la respuesta de la misma pregunta realizada.

Para cada elemento (pregunta almacenada) del listado de la base de datos, realizarán los siguientes pasos.

Segundo: Se procesará LCS del elemento contra Dato ingresado.

En este paso, se procesa el elemento aplicando el algoritmo LCS, una vez que se ha procesado se toma la cadena resultante del proceso.

Tercero: Se procesará la distancia Levenshtein a la cadena LCS contra Dato ingresado.

Se toma la cadena resultante de LCS y la cadena ingresada para aplicar la distancia Levenshtein, una vez que se culmina el procesamiento se toma el valor de la distancia, si ésta es menor a la mitad de la longitud de la pregunta ingresada, esta es almacenada en un diccionario. En caso

que ninguna pregunta cumpla las pasadas condiciones significa que no se tiene esa pregunta en la base de datos.

Cuarto: Ubicando respuesta.

Como se ha mencionado, la lista de las posibles respuestas (con sus respectivos valores) se tienen almacenadas en un diccionario, lo que se hace es ordenar la lista y tomar el de menor valor, para luego mostrarlo.

En el próximo apartado, se muestra el código fuente que sustenta y grafica la lógica mostrada en esta explicación, clarificando de mejor manera este método.

5.4. Implementar en lenguaje de programación el método de procesamiento de consultas.

Se ha venido mostrando en una secuencia lógica como se ha trabajado el proyecto de desarrollo de la herramienta, en este punto se muestra el código fuente, considerado el corazón de nuestra herramienta que vendría a ser el motor de búsqueda que se encuentra en la Lógica de negocio. A continuación, se muestra el código fuentes en java:

5.4.1. Analizador de Consulta.

Cuadro 15 : Descripción Código Analizador de Consulta.

Analizador.java

```
package Logica;
import Bean.BeanProblema;
import Bean.BeanSolucion;
import java.util.Arrays;
import java.util.Date;
import java.util.Hashtable;
import java.util.List;
import java.util.Map;

public class Analizador {
static String enviar="";
    public static  BeanProblema getRespuesta(String pregunta) throws Exception {

        String respuesta = "";
        LogicaProblema logicaProblema = new LogicaProblema();
```

```
LogicaSolucion logicaSolucion = new LogicaSolucion();

//Primero se obtiene las preguntas de la base de datos
Hashtable<BeanProblema, Integer> posiblesRespuestas = new Hashtable<BeanProblema, Integer>();

String Inicio_Proceso = "Proceso iniciado";
String Fin_Proceso = "Fin del proceso";
String Formato_DosDigitos = "00";
Date fechaInicio, fechaFin;

System.out.println(Inicio_Proceso + "</br>");
enviar=enviar +"" +Inicio_Proceso+"</br>";
fechaInicio = new Date();
long time_start, time_end;
time_start = System.currentTimeMillis();

List<BeanProblema> preguntasEnLaBaseDeDatos = logicaProblema.listar();
```

```
//Luego se revisa que cada una de esas preguntas sea igual a las de la base de datos si es así se devuelve la misma pregunta
        for (int i = 0; i < preguntasEnLaBaseDeDatos.size(); i++) {
            if (pregunta.equals(preguntasEnLaBaseDeDatos.get(i).getDescripcion())) {
                List<BeanSolucion> solucionEnLaBaseDeDatos = logicaSolucion.listar( preguntasEnLaBaseDeDatos.get( i ).getIdproblema( ));
                respuesta = solucionEnLaBaseDeDatos.get(0).getDescripcion();
                time_end = System.currentTimeMillis();
                //System.out.println("Tiempo transcurrido:" + (time_end - time_start) / 1000 % 60 + " milliseconds \t");
                enviar=enviar+"Tiempo transcurrido:" + (time_end - time_start) / 1000 % 60 + " segundos </br>";
                return preguntasEnLaBaseDeDatos.get(i);
            }
        }
        for (int i = 0; i < preguntasEnLaBaseDeDatos.size(); i++) {
            //Aquí se obtiene la LCS
            String lcs = Util.lcsIgnoreCaps(pregunta, preguntasEnLaBaseDeDatos.get(i).getDescripcion());
            //Aquí se obtiene la distancia de levenshtein
            LogicaLevenshtein lvs = new LogicaLevenshtein(pregunta, lcs);
```

```
            int dist = lvs.getSimilarity();

            //Aquí está donde se compara la distancia si la distancia es mayor de la mitad de la longitud de la pregunta entonces se descarta
            if (dist < (pregunta.length() * 0.5f)) {
                enviar=enviar+ "lcs: " + lcs +"\t";
                //System.out.println("lcs  " + lcs);
                enviar=enviar +"Pregunta a comparar: " +preguntasEnLaBaseDeDatos.get(i).getDescripcion() + " distancia: " + dist +"</br>";
                //en caso de que la distancia sea menor se va almacenando en un diccionario:  pregunta + distancia
                posiblesRespuestas.put(preguntasEnLaBaseDeDatos.get(i), dist);
            }
        }
        //En caso ninguna pregunta cumpla las pasadas condiciones significa que no se tiene esa pregunta en la base de datos
        if (posiblesRespuestas.isEmpty()) {
            return null;
        }
```

```
//Luego se obtiene un array con todas las distancias
Integer[] distancias = new Integer[posiblesRespuestas.size()];
posiblesRespuestas.values().toArray(distancias);
//Se ordena de forma descendiente y así se tiene la menor distancia primero
Arrays.sort(distancias);
System.out.println(Arrays.toString(distancias));
int index = 0;
while (respuesta.equals("")) {
    for (Map.Entry entry : posiblesRespuestas.entrySet()) {
        if (distancias[index] == entry.getValue()) {
            if (((BeanProblema) entry.getKey()).getDescripcion().length() >= (pregunta.length() * 1.1)) {
                if (index < distancias.length - 1) {
                    enviar=enviar+ "index: " + (index + 1) +"</br>";
                    //System.out.println("index " + (index + 1));
                    index++;
                    continue;
                } else {
                    return null;
```

```
                }
            }
             enviar=enviar+ "ID:Problema (Respuesta): " + ((BeanProblema) entry.getKey()).getIdproblema()
+"</br>";
            //Aquí se obtiene las soluciones para la pregunta ya que si se llega hasta esta parte del código significa
            //que una de las pregunta logro pasar las verificaciones anteriores
            List<BeanSolucion> solucionesPosible = logicaSolucion.listar(((BeanProblema)
entry.getKey()).getIdproblema());
            //Se obtiene la descripción de la solución.
            String solucion = solucionesPosible.get(0).getDescripcion();
            respuesta = solucion;
            //Aquí devuelve el Objeto problema que contiene la descripcion y el id de la pregunta
            //que está en la base de datos y que es similar a la pregunta que se ingresó.
            time_end = System.currentTimeMillis();
            enviar=enviar +"Tiempo transcurrido:" + (time_end - time_start) / 1000 % 60 + " segundos</br>";
            return ((BeanProblema) entry.getKey());
        }
    }
```

```
        }
        time_end = System.currentTimeMillis();
        System.out.println("Tiempo transcurrido:" + (time_end - time_start) / 1000 % 60 + " segundos");
        return null;
    }
    public static String getEnviarDatos(){
        return enviar;
    }
}
```

Fuente: Diseño propio.

5.4.2. Selección de Sub conjunto de datos

Cuadro 16 : Descripción Código Sub Conjunto de Datos.

Util.java

```
package Logica;


public class Util {
	public static String lcs(String str1, String str2) {
	int l1 = str1.length();
	int l2 = str2.length();
	int[ ][ ] arr = new int[l1 + 1][l2 + 1];
	for (int i = l1 - 1; i >= 0; i--) {
		for (int j = l2 - 1; j >= 0; j--) {
			if (str1.charAt(i) == str2.charAt(j))
				arr[i][j] = arr[i + 1][j + 1] + 1;
			else
				arr[i][j] = Math.max(arr[i + 1][j], arr[i][j + 1]);
		}
	}
```

```
        int i = 0, j = 0;
        StringBuffer sb = new StringBuffer();
        while (i < l1 && j < l2) {
                if (str1.charAt(i) == str2.charAt(j)) {
                        sb.append(str1.charAt(i));
                        i++;
                        j++;
                } else if (arr[i + 1][j] >= arr[i][j + 1])
                        i++;
                else
                        j++;
        }
        return sb.toString();
}
        public static String lcsIgnoreCaps(String str1, String str2) {
        int l1 = str1.length();
        int l2 = str2.length();
```

```
int[][] arr = new int[l1 + 1][l2 + 1];
for (int i = l1 - 1; i >= 0; i--) {
        for (int j = l2 - 1; j >= 0; j--) {
                if (Character.toLowerCase(str1.charAt(i)) == Character.toLowerCase(str2.charAt(j)))
                 {  arr[i][j] = arr[i + 1][j + 1] + 1  }
                 Else {  arr[i][j] = Math.max(arr[i + 1][j], arr[i][j + 1])  } }
        }
        int i = 0, j = 0;
        StringBuffer sb = new StringBuffer();
        while (i < l1 && j < l2) {
                if (Character.toLowerCase(str1.charAt(i)) == Character.toLowerCase(str2.charAt(j))) {
                        sb.append(str1.charAt(i));
                        i++;
                        j++;
                } else if (arr[i + 1][j] >= arr[i][j + 1])
                        i++;
                else
                        j++;
```

```
        }
        return sb.toString();
}
public static int longestSubstr(String first, String second) {
        if (first == null || second == null || first.length() == 0 || second.length() == 0) {
                return 0;
        }
        int maxLen = 0;
        int fl = first.length();
        int sl = second.length();
        int[][] table = new int[fl][sl];
        for (int i = 0; i < fl; i++) {
                for (int j = 0; j < sl; j++) {
                        if (first.charAt(i) == second.charAt(j)) {
                                if (i == 0 || j == 0) {
                                        table[i][j] = 1;
                                } else {
                                        table[i][j] = table[i - 1][j - 1] + 1;
```

```
                                    }
                                    if (table[i][j] > maxLen) {
                                            maxLen = table[i][j];
                                    }
                            }
                    }
            }
            return maxLen;
    }
}
```

Fuente: Diseño propio.

5.4.3. Lógica Levenshtein

Cuadro 17 : Descripción Código Lógica de Levenshtein.

LogicaLevenshtein.java
package Logica; public class LogicaLevenshtein

```
{
    private String compOne;
    private String compTwo;
    private int[][] matrix;
    private Boolean calculated = false;
     public LogicaLevenshtein(String one, String two)
    {
        compOne = one;
        compTwo = two;
    }
     public int getSimilarity()
    {
        if (!calculated)
        {
            setupMatrix();
        }
    return matrix[compOne.length()][compTwo.length()];
    }
```

```
 public int[][] getMatrix()
{
    setupMatrix();
    return matrix;
}
 private void setupMatrix()
{
    matrix = new int[compOne.length()+1][compTwo.length()+1];
System.out.println("MAtriz de comparacion");
    for (int i = 0; i <= compOne.length(); i++)
    {
        matrix[i][0] = i;
    }
     for (int j = 0; j <= compTwo.length(); j++)
    {
        matrix[0][j] = j;
    }
     for (int i = 1; i < matrix.length; i++)
```

```
{
    for (int j = 1; j < matrix[i].length; j++)
    {
        if (compOne.charAt(i-1) == compTwo.charAt(j-1))
        { matrix[i][j] = matrix[i-1][j-1]  }
        else
        {
            int minimum = Integer.MAX_VALUE;
            if ((matrix[i-1][j])+1 < minimum)
            {  minimum = (matrix[i-1][j])+1 }; //Eliminar
              if ((matrix[i][j-1])+1 < minimum)
            {  minimum = (matrix[i][j-1])+1 } ; //insertar
            if ((matrix[i-1][j-1])+1 < minimum)
            { minimum = (matrix[i-1][j-1])+1 }; //Sustituir
             matrix[i][j] = minimum;
        }
   System.out.print(matrix[i][j]+" ");
    }
```

```
            System.out.println();
        }
         System.out.println("Fin de matriz");
        calculated = true;
    }
   private void displayMatrix()
    {
        System.out.println("  "+compOne);
        for (int y = 0; y <= compTwo.length(); y++)
        {
            if (y-1 < 0) System.out.print(" "); else System.out.print(compTwo.charAt(y-1));
            for (int x = 0; x <= compOne.length(); x++)
            { System.out.print(matrix[x][y]) } ;
            System.out.println();
        }
    }
}
```

Fuente: Diseño propio.

CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones.

1. Se seleccionó la técnica a implementar en el desarrollo de la herramienta SIPLENAST (Sistema de procesamiento de lenguaje natural de Soporte técnico). Después de haber revisado literatura relacionada se determinó que la técnica basada en la implementación del algoritmo de Levenshtein es una de las más efectivas, adicionalmente que se cuenta con basta documentación que permite su fácil implementación a comparación de otras técnicas similares.
2. Se diseñó la herramienta presentada bajo la metodología RUP y el Lenguaje Unificado de Modelado (UML), conformando la metodología orientada a objetos para el desarrollo de software.
3. Se desarrolló esta herramienta considerando estándares de desarrollo propias, el lenguaje de programación usado para el desarrollo del software ha sido Java teniendo como IDE Netbeans y como motor de base de datos PostgreSQL que son herramientas de licenciamiento libre.
4. Se determinó que la técnica de implementación de Levenshtein tuvo que ser complementada con otro algoritmo el LCS (Longest common subsequence) haciendo una versión mejorada de nuestro motor de búsqueda.
5. Se evaluaron los resultados en base a las variables operacionalización consideradas, como son el Tiempo y la Precisión, las cuales brindaron

resultados alentadores con respecto al motor de búsqueda de la herramienta SIPLENAST ya que, con respecto al tiempo, el promedio de respuesta es de 112.93 ms, siendo menor o mayor dependiendo al acercamiento que tenga del texto ingresado como consulta con la pregunta almacenada; por otro lado con respecto a la precisión, se tiene un 93.33% de precisión, es decir casi la totalidad de las consultas fueron acertadas con respecto a la respuesta esperada.

6.2. Recomendaciones.

1. Se recomienda seguir investigando y difundiendo el estudio acerca del procesamiento de lenguaje natural y los sistemas de búsqueda de respuesta, ya que son áreas poco estudiadas a nivel de región y país.
2. El estudio de esta rama de la inteligencia artificial se viene valorizando cada vez más, desde el punto de vista social se recomienda su aplicación en talleres de esta carrera, ya que les servirá a los egresados, llevar esos conocimientos al ámbito laboral donde podrá aplicar estas nuevas tecnologías, los resultados se verán reflejadas en un futuro en el prestigio de la universidad.
3. Se recomienda realizar mayores investigaciones de esta rama ya que permitirá encontrar nuevas soluciones a problemas cotidianos como especializados, mejorar las que se tienen a disposición y encontrar usos sofisticados a diversas tareas del ámbito laboral y de investigación.

4. Se recomienda que para el uso de SIPLENAST en la búsqueda de respuestas no utilizar frases demasiado extensas porque la recuperación de la información ralentiza el motor de búsqueda y pierde precisión.
5. Para el desarrollo de este tipo de herramientas se recomienda hacerlas en java por dos motivos: el primero porque las diversas técnicas y las aplicaciones de demostración con los que se cuenta vienen para este lenguaje de programación y el segundo porque existe basta documentación sobre estas técnicas para implementar en este lenguaje.
6. Para poder tener una mejor comprensión de lo propuesto en punto anterior es recomendable tener conocimiento del idioma inglés a nivel intermedio avanzado ya que la mayoría de información que se encuentra al respecto es en este idioma.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Referencias

Allen, J. (1995). *Natural language understanding.* Redwood City: The Benjamin / Cummings Publishing Company.

Amón, I., & Jiménez, C. (2012). *Funciones de Similitud sobre Cadenas de Texto: Una Comparación Basada en la Naturaleza de los Datos.* Bogotá, Colombia: Universidad Pontificia Bolivariana.

Ariel Domínguez, M. (04 de 06 de 2012). Tesis doctoral en ciencias de la computación. *Optimization of automata for natural laguage processing dependency parsing*. Córdoba, Córdoba, España: FAMAF-UNC-2012.

Bach, E. W. (1989). *Informal lectures on formal semantics.* New York: State University of New York Press.

Biblioteca Universidad Católica Santo Toribio de Mogrovejo. (2006). Aplicaciones de la Inteligencia artificial. *Aplicaciones de la Inteligencia artificial*. Chiclayo, Lambayeque, Perú: Biblioteca Universidad Católica Santo Toribio de Mogrovejo.

Bon, J. v., de Jong, A., Kolthof, A., Pieper, M., Tjassing, R., van der Veen, A., & Verheijen, T. (2008). *Gestión de Servicios TI basado en ITIL® V3 - Guia de Bolsillo.* Van Haren Publishing, Zaltbommel.

Cabeza Barrios, J. E., & Torres, O. R. (1989). *Enciclopedia Estudiantil Educar: Gramática Española.* Bogotá: Editorial Educar Cultural Recreativa.

Carnegie Mellon University - Computer Science departament. (28 de Febrero de 2015). *https://www.csd.cs.cmu.edu/.* Obtenido de https://www.csd.cs.cmu.edu/: http://www.cs.cmu.edu/afs/cs/academic/class/15451-s15/LectureNotes/lecture04.pdf

Chandrasekaran, S., & DiMascio, C. (30 de 05 de 2014). *http://www.ibm.com/.* Obtenido de http://www.ibm.com/: http://www.ibm.com/developerworks/ssa/cloud/library/cl-watson-films-bluemix-app/cl-watson-films-bluemix-app-pdf.pdf

Chierchia, G. (1990). *Meaning and grammar.* Cambridge: MIT Press.

Díaz, D. T. (15 de Junio de 2009). Tesis Doctoral. *Sistemas de clasificación de preguntas basados en corpus para búsquedas de respuestas*. Alicante, Alicante, España: Universidad de Alicante.

F.J., M. M., & J.L., R. R. (2012). Procesamiento del lenguaje natural. *Inteligencia Artifical II* (pág. 8). Sevilla: Dpto. Ciencias de la Computación e Inteligencia Artificial Universidad de Sevilla.

Fernández, D. (2015). *MERA: Musical Entities Reconciliation Architecture.* Oviedo, Asturias, España: Universidad de Oviedo.

Gamallo Otero, P., & García Gonzalez, M. (2012). *Técnicas de Procesamiento del Lenguaje Natural en la recuperación de información*. Santiago de Compostela, La Coruña , España: Universidad de Santiago de Compostela, España.

Gutiérrez-Artacho, J. (2014). *Recursos y herramientas lingüísticos para los sistemas de búsquedas de respuestas monolingües y multilingües.* Tesis Doctoral, Universidad de Granada, Programa Oficial de postgrado en Estudios avanzados de traducción e interpretación, Granada. Obtenido de http://digibug.ugr.es/bitstream/10481/40371/1/24934550.pdf

Hardy, T. (2006). IA: Inteligencia artificial. *E-Libro - Polis revista de la Universidad Bolivariana*, 6-10.

Hernández, M., & Gómez, J. (Julio de 2013). Aplicaciones de Procesamiento de Lenguaje Natural. *Revista Politécnica - Escuela Politécnica Nacional Ecuador, Vol 32*(1), 87-96.

Iván, A., & Claudia, J. (2010). *The Open University*. (U. P. Bolivariana, Ed.) Obtenido de https://core.ac.uk/download/pdf/11052347.pdf

Jaramillo, S., & Londoño, J. M. (2014). Búsqueda de documentos basada en el uso de índices ontológicos creados con MapReduce. *Ciencia e Ingeniería Neogranadina*, 57 - 75.

Kay, M. (2004). *The Oxford Handbook of Computational.* Oxford: Oxford University Press.

Magnini, B., Lavelli, A., Fabien, G., Cabrio, E., Cojan, J., & Palmero Aprosio, A. (2005). Open Domain Question Answering: Techniques, Systems and Evaluation. *RANLP*.

Martínez-Barco, P., Vicedo, J. L., Saquete, E., & Tomás, D. (2009). Grupo de Procesamiento del Lenguaje y Sistemas de Información. *Sistemas de Pregunta-Respuesta*. Alicante, Alicante, España: Universidad de Alicante.

McEnery, T. (1992). *Computational linguistics: a handbook & toolbox for natural language processing.* Wilmslow: Sigma Press.

Mesones Barrón, c. E. (2006). Tesis . *COMPRENSIÓN Y GENERACIÓN DE LENGUAJE NATURAL EN UN SISTEMA DE DIÁLOGO USANDO INTELIGENCIA ARTIFICIAL PARA SERVICIO TELEFÓNICOS DE INFORMACIÓN DE CINES*. Lima, Lima, Perú: UPC.

Meya, M., & Huber, W. (1986). *Lingüística computacional.* Barcelona: Teide.

Olvera-Lobo, M.-D., & Robinson-García, N. (2009). Tratamiento lingüístico de las preguntas en español para su clasificación en los sistemas de búsqueda de respuestas. *El profesional de la información*, 180-187.

Parcero, E. (25 de Setiembre de 2012). *Implementación de Técnicas de String Matching y selección semántica aproximada en un motor de normalización terminológica*. Obtenido de Universitat Politècnica de Valéncia: https://riunet.upv.es/bitstream/handle/10251/17464/Memoria.pdf?sequence=1

Pedraza-Jimenez, R., & Vallez, M. (2007). *http://www.hipertext.net*. Obtenido de https://www.upf.edu/hipertextnet/numero-5/pln.html#procesamiento-linguistico-lenguaje-natural

Peña Ayalla, A. (2006). *Lenguaje Natural: Descripción de las etapas para su tratamiento.* DF México: INSTITUTO POLITÉCNICO NACIONAL.

Quevedo, M. Á., Rosique, P., Ruiz, F., & Aldeguer, R. (1999). Fundamentos de la Inteligencia Artificial. *EBSCO*, 1 - 5.

Reichgelt, H. (1991). *Knowledge representation: an AI perspective.* Norwood: Ablex.

Rodríguez, J. (5 de 10 de 2012). *Blog procesamiento Lenguaje Natural*. Obtenido de http://pdln.blogspot.pe/2012/10/distancia-minima-de-edicion.html

Ruiz, J., Juárez, R., Cervantes, J., & Trueba, A. (Setiembre de 2015). *www.uaemex.mx.* (U. D. Colombia., Ed.) Obtenido de Universidad Autónoma del Estado de México: http://hdl.handle.net/20.500.11799/40636

Sanderson, M. (2000). Retrieving with good sense. En M. Sanderson, *Retrieving with good sense* (págs. 49-69). Sheffield: The University of Sheffield.

SHRDLU, T. W. (1970). SHRDLU . EEUU.

Sosa, E. (1997). Procesamiento del lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones. *El profesional de la información*, 26-29.

Vicedo González, J. L. (25 de Abril de 2002). Tesis doctoral. *SEMQA: un modelo semántico aplicado a los sistemas de búsqueda de respuestas*. Alicante, Alicante, España: Universidad de Alicante.

Vila, K. (8 de Octubre de 2010). Búsqueda de respuestas en dominios restringidos: aplicación sobre el dominio agrícola. *Tesis doctorales*. Alicante, Alicante, España: Dpto de lenguajes y sistemas informáticos.

Vila, K., Mazón, J.-N., & Ferrández, A. (2012). Creación automática de sistemas de búsqueda de respuestas en dominios restringidos. *El profesional de la información*, 16-26.

Wooten, B., & Wooten, R. J. (2001). *Building & Managing a World Class IT Help Desk.* New York, London: McGraw-Hill Professional.

Zajac, R. (2001). *Towards ontological question answering.* Toulouse, France: Association for Computational Linguistics Stroudsburg, PA, USA 2001.