

**REVISIÓN SISTEMÁTICA SOBRE HERRAMIENTAS PARA LA EXTRACCIÓN  
DE CASOS DE USO MEDIANTE PROCESAMIENTO DE LENGUAJE NATURAL  
A PARTIR DE LA DOCUMENTACIÓN RESULTANTE DE LA ELICITACIÓN DE  
REQUERIMIENTOS**

**CARLOS JAVIER ANCHICO PEÑA  
RAFAEL ANDRES BELALCAZAR**



**FUNDACIÓN  
UNIVERSITARIA DE POPAYÁN**

**FUNDACION UNIVERSITARIA DE POPAYAN  
FACULTAD DE INGENIERÍAS  
PROGRAMA DE INGENIERIA DE SISTEMAS  
POPAYAN CAUCA  
2018**

**REVISIÓN SISTEMÁTICA SOBRE HERRAMIENTAS PARA LA EXTRACCIÓN  
DE CASOS DE USO MEDIANTE PROCESAMIENTO DE LENGUAJE NATURAL  
A PARTIR DE LA DOCUMENTACIÓN RESULTANTE DE LA ELICITACIÓN DE  
REQUERIMIENTOS**

**CARLOS JAVIER ANCHICO PEÑA  
RAFAEL ANDRES BELALCAZAR**

**PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO DE  
SISTEMAS**

**DIRECTOR  
DR. GUSTAVO ANDRÉS URIBE GÓMEZ**



**FUNDACIÓN  
UNIVERSITARIA DE POPAYÁN**

**FUNDACION UNIVERSITARIA DE POPAYAN  
FACULTAD DE INGENIERÍAS  
PROGRAMA DE INGENIERIA DE SISTEMAS  
POPAYAN CAUCA  
2018**

Nota de aceptación:

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

--- Pagina dedicatoria

--- Pagina de agradecimientos

# 1 CONTENIDO

	pág.
2 RESUMEN .....	8
3 ABSTRACT.....	9
4 INTRODUCCIÓN.....	10
5 CAPÍTULO 1. DEFINICIÓN DEL PROBLEMA.....	11
1.1. ESPECIFICACIÓN DEL PROBLEMA .....	11
1.2. JUSTIFICACIÓN .....	12
1.3. OBJETIVOS.....	13
1.3.1. Objetivo general .....	13
1.3.2. Objetivos específicos.....	13
1.4. METODOLOGÍA.....	14
1.5. MARCO TEÓRICO .....	17
1.6. ORGANIZACIÓN DEL DOCUMENTO .....	21
6 CAPÍTULO 2. REVISIÓN SISTEMÁTICA DE TRABAJOS RELACIONADOS .....	22
6.1 DE LOS ARTÍCULOS INCLUIDOS ES POSIBLE IDENTIFICAR LOS SIGUIENTES .....	22
6.2 Características .....	22
7 CAPÍTULO 3. ANÁLISIS DE METODOLOGÍAS DE PROCESAMIENTO DE LENGUAJE NATURAL PARA EXTRACCIÓN DE CASOS DE USO .....	28
3.1. PROCESAMIENTO DE LENGUAJE NATURAL .....	28
3.1.1. Tipos de procesamiento de lenguaje natural .....	28
3.1.2. Niveles del procesamiento de lenguaje natural por texto .....	29
3.1.3. Técnicas del procesamiento de lenguaje natural por texto .....	30
3.1.4. Problemas en el procesamiento de lenguaje natural .....	33
3.2. HERRAMIENTAS DE PROCESAMIENTO DE LENGUAJE NATURAL .....	35
3.2.1. Apache OpenNLP.....	35
3.2.2. NLTK .....	36
3.2.3. Freeling.....	36
3.3. COMPARACIÓN DE LAS HERRAMIENTAS DE PROCESAMIENTO DE LENGUAJE NATURAL.....	38
8 CAPÍTULO 4. HERRAMIENTA PARA LA ASISTENCIA EN LA EXTRACCIÓN DE CASOS DE USO .....	40

9	CAPÍTULO 5. EVALUACIÓN DE LA HERRAMIENTA DESARROLLADA .....	43
10	CAPÍTULO 6. CONCLUSIONES Y TRABAJOS FUTUROS .....	44
11	BIBLIOGRAFÍA.....	45

## 2 RESUMEN

En el presente trabajo se realiza un estudio de los métodos para la extracción de casos de usos mediante procesamiento de lenguaje natural a partir de la elicitación de requerimientos. Dichos métodos buscan reducir el tiempo en la extracción de casos de uso en un proceso de elicitación de requerimientos. Tras analizar las metodologías para el procesamiento del lenguaje natural, que permitan la extracción de casos de uso a partir de una entrevista al usuario, se pudo evidenciar que los métodos propuestos hasta la fecha no han sido implementados en una herramienta.

### ***Palabras clave***

Elicitación de requerimientos, procesamiento de lenguaje natural, extracción de casos de uso, ingeniería de requisitos.



### 3 ABSTRACT

In the present work a study of the methods for the extraction of cases of uses by means of processing of natural language is made from the elicitation of requirements. These methods seek to reduce the time in the extraction of use cases in a process of elicitation of requirements. After analyzing the methodologies for natural language processing, which allow the extraction of use cases from a user interview, it was possible to demonstrate that the methods proposed to date have not been implemented in a tool.

#### ***Key Words***

Elicitation of requirements, natural language processing, extraction of use cases, requirements engineering.

## 4 INTRODUCCIÓN

Según la Ingeniería del Software, el punto de partida para un proceso de desarrollo de software es la elicitación de requerimientos [Pressman, 2005], en dicho proceso se acuerda junto con el cliente un listado de características a satisfacer con el sistema informático [Ian Sommerville, 2006]. Como salida de este proceso se formaliza un documento de Especificación de Requisitos Software que debe ser completo y aceptado por los interesados. Dicho documento usualmente se encuentra especificado en lenguaje natural y es el recurso requerido para la obtención de los casos de uso del sistema. Usualmente el proceso de interpretación del documento de Especificación de Requisitos es realizado de manera manual, lo que toma un tiempo considerable dentro del desarrollo del software.

En la actualidad existen diversas herramientas para el apoyo en los procesos asociados al desarrollo de software, un ejemplo son las herramientas denominadas herramientas de ingeniería de software asistida por computadora (CASE, Computer-Aided Software Engineering), automatizando procesos como la generación de código a partir de diagramas UML [Zapata, 2007]. Sin embargo, previamente a este documento no existe un análisis de las herramientas para el soporte en el proceso de construcción automatizada de casos de uso. El presente artículo responde a esta necesidad estudiando, mediante una revisión sistemática, los aportes científicos en relación con las herramientas software de asistencia al proceso de generación casos de uso a partir de la documentación obtenida en el proceso de elicitación de requerimientos. A continuación se describirá el procedimiento metodológico utilizado para la revisión sistemática, posteriormente se mostrarán los resultados obtenidos incluyendo una caracterización de las propuestas actuales y finalmente se presentarán las conclusiones sobre el tema.

## 5 CAPÍTULO 1. DEFINICIÓN DEL PROBLEMA

### 1.1. ESPECIFICACIÓN DEL PROBLEMA

El planteamiento del problema surge a partir de encontrar una herramienta que les permita a los analistas de sistemas poder realizar los diagramas de casos de uso de forma rápida y eficiente, vale aclarar que no se trata de eliminar la tarea del analista más bien esta herramienta le puede asistir en la generación de los diagramas y así tener procesos con mayor iteración con el desarrollo.

Basado en esta problemática surge la pregunta, ¿Cómo reducir el tiempo en la extracción de casos de uso en un proceso de elicitación de requerimientos?

Los casos de uso son una técnica para especificar el comportamiento de un sistema: Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. Todo sistema de software ofrece a su entorno – aquellos que lo usan– una serie de servicios. Un caso de uso es una forma de expresar cómo alguien o algo externo a un sistema lo usa. Cuando decimos “alguien o algo” hacemos referencia a que los sistemas son usados no sólo por personas, sino también por otros sistemas de hardware y software. Por ejemplo, un sistema de ventas, si pretende tener éxito, debe ofrecer un servicio para ingresar un nuevo pedido de un cliente. Cuando un usuario accede a este servicio, podemos decir que está “ejecutando” el caso de uso ingresando pedido (Ceria, 2001).

## **1.2. JUSTIFICACIÓN**

En el proyecto de investigación se realiza un estudio de los métodos o técnicas que se usan actualmente para el procesamiento de lenguaje natural, esto con el fin de proponer una herramienta que nos permita la extracción de casos de usos mediante procesamiento de lenguaje natural a partir de la elicitación de requerimientos. Esto con el fin de reducir el tiempo en la extracción de casos de uso en un proceso de elicitación de requerimientos.

El propósito de la investigación es proponer una herramienta que nos permita asistir en la obtención de los casos de uso de forma automatizada a partir de la captura de requerimientos, cuyo suministro sea la entrevista o requerimientos obtenidos del stakeholder.

### **1.3. OBJETIVOS**

#### **1.3.1. Objetivo general**

Proponer una herramienta para facilitar la extracción de casos de uso en la elicitación de requerimientos.

#### **1.3.2. Objetivos específicos**

- Analizar las metodologías para el procesamiento del lenguaje natural, que permitan la extracción de casos de uso a partir de una entrevista al usuario.
- Desarrollar una herramienta que permita asistir en la elicitación de requerimientos, mediante herramientas de lenguaje natural.
- Evaluar la reducción de tiempo y la completitud en la identificación de casos de uso dentro del desarrollo de software.

## 1.4. METODOLOGÍA

Para la presente revisión sistemática se ha seguido la metodología descrita en [Keele, 2007]. Las búsquedas de documentos científicos se realizaron sobre las siguientes bases de datos bibliográficas:

- IEEExplorer
- ACM Digital library
- Google scholar (scholar.google.com)
- SpringerLink

Las cadenas de búsqueda utilizadas para estos motores de búsqueda fueron:

1. ("ingeniería de requerimientos" OR "elicitación de requerimientos" OR "elicitación de requisitos") AND ("lenguaje natural" OR "NLP" OR "PLN")
2. ("ingeniería de requerimientos" OR "elicitación de requerimientos" OR "elicitación de requisitos") AND ("OpenNLP" OR "ClearTK" OR "Apache NLP" OR "CoreNLP" OR "UIMA" OR "Word net")
3. ("OpenNLP" OR "ClearTK" OR "Apache NLP" OR "CoreNLP" OR "UIMA" OR "Word net") AND "casos de uso" AND ("desarrollo de software" OR "implementación de software")
4. "software development" AND ("requirements engineering" OR "requirements elicitation") AND ("natural language processing" OR "NLP") NOT ("collaborative software" OR "analysis of requirements" OR "data mining" OR "aspect-oriented" OR "transformation")
5. "software development" AND ("requirements engineering" OR "requirements elicitation") AND ("OpenNLP" OR "ClearTK" OR "Apache NLP" OR "CoreNLP" OR "UIMA" OR "wordnet") NOT ("ontology-based" OR "software testing")
6. "software development" AND ("use cases" OR "use case") AND ("OpenNLP" OR "ClearTK" OR "Apache NLP" OR "CoreNLP" OR "UIMA" OR "wordnet") NOT ("ontology-based" OR "software testing" OR "ontology driven" OR "Unstructured Information Management Architecture" OR "Model-driven")

Las cadenas de búsqueda 1 y 4 indagan en general acerca de los trabajos que tratan en intervienen los procesos de elicitación de requerimientos mediante el procesamiento de lenguaje. Para la cadena número 4,5 y 6 se tuvo en cuenta algunos de los criterios de exclusión, incluyendo el operador lógico NOT. Para la construcción de las cadenas 2 y 5 se consideraron las herramientas de PLN más conocidas y que hubiesen sido aplicadas en el proceso de elicitación de requerimientos. Dichas herramientas también se tuvieron en cuenta en la cadenas 3 y 6, pero en este caso se indago por su uso en la construcción de casos de uso.

### **Criterios de Inclusión.**

Se incluyeron los artículos y trabajos de investigación que cumplan todas las siguientes características:

1. Enfocadas en el procesamiento del lenguaje natural en las primeras etapas del desarrollo de un sistema software.
2. Utilizan técnicas o herramientas para el procesamiento del lenguaje natural como: Clear NLP, Open NLP, Apache NLP, Core NLP, Clear TK, UIMA, herramientas Upper CASE, herramientas CASE en la ingeniería del software.
3. Artículos relacionados con el modelado o extracción de conocimientos mediante el procesamiento del lenguaje natural en la ingeniería del software.

### **Criterios de Exclusión.**

Se excluyeron trabajos se cumpliesen al menos una de las siguientes características:

1. Anteriores al año 2012.
2. Catalogados como ontologías o basados en ontologías.
3. Enfoque de estudio limitado a una organización.
4. Que no tengan relación con el procesamiento del lenguaje natural mediante técnicas o herramientas en las primeras etapas del desarrollo de un sistema software.
5. Que correspondan a software colaborativo, basado en aspectos o un algoritmo de transformación.
6. Artículos dedicados a la elaboración de revisiones y síntesis de estudios anteriores.

## Resultados

La figura 1 evidencia el número de publicaciones encontradas luego de usar las cadenas de búsqueda para cada una de las bases de datos bibliográficas.

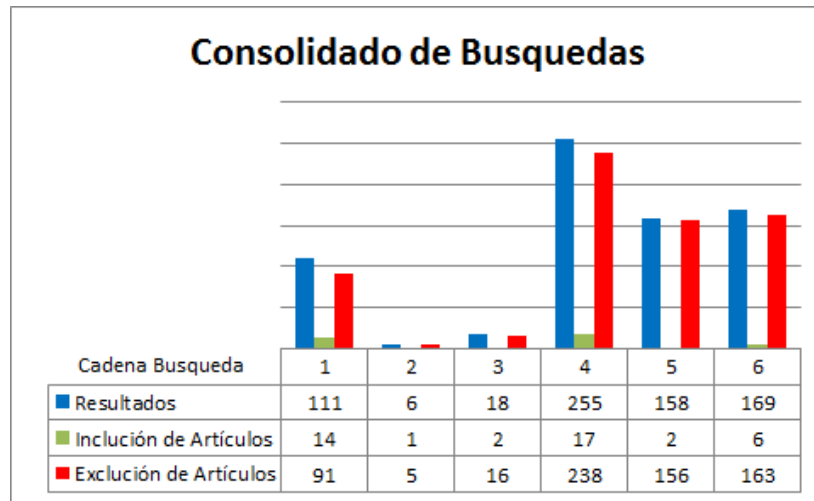


Figura 1. Publicaciones resultado de las cadenas de búsqueda

La figura 2 muestra el número de artículos incluidos por año de publicación, donde podemos evidenciar claramente que esta temática está en continuo incremento siendo una tendencia actual de investigación.

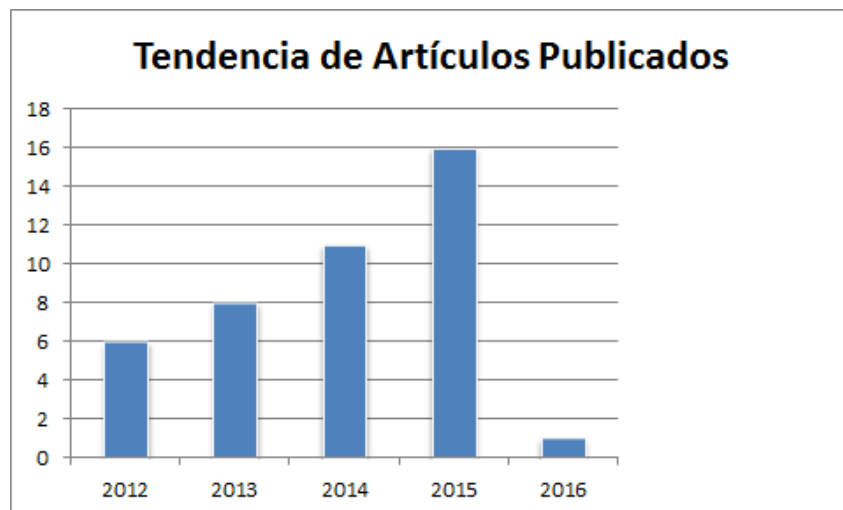


Figura 2. Publicaciones por año de publicación



## **1.5. MARCO TEÓRICO**

### **Procesamiento de Lenguaje Natural (PLN)**

El procesamiento de lenguaje natural es una sub disciplina de la Inteligencia Artificial y la rama ingenieril de la lingüística computacional. El Procesamiento de Lenguaje Natural se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas o entre personas y máquinas por medio de lenguajes naturales. (Salas, 2011)

### **Ingeniería de Requisitos**

La Ingeniería de Requisitos es la primera fase en el desarrollo de software, cuyas actividades centrales se orientan a descubrir, capturar, analizar y especificar los requisitos que el futuro software necesita satisfacer para que se considere de calidad.

Estas actividades se deben llevar a cabo de manera iterativa e incremental y priorizando por su nivel de criticidad la ejecución de los procesos de educación de requisitos y análisis de requisitos. Estos procesos involucran la identificación, adquisición y procesamiento de lenguaje natural, para su posterior representación y/o especificación en un lenguaje de modelado (Li, 2003).

Según Robertson y Robertson (Robertson S. y., 2008) en el proceso general de ingeniería de requisitos, se desarrollan ciertas actividades, como: entendimiento del dominio de aplicación, captura y clasificación de requisitos, establecimiento de prioridades, resolución de conflictos y negociación de los requisitos del sistema. Estas actividades se consideran clave y críticas, principalmente por la acción de comunicación analista-interesado, que está orientada hacia comprender y recuperar la información esencial y relevante acerca del dominio, que se convertirá en la base de los requisitos y la cual se extrae de los interesados (cliente, usuario final, experto del dominio).

## **Precisión**

La precisión es el ratio entre el número de documentos relevantes recuperados entre el número de documentos recuperados. Acorde a la definición se tiene la siguiente expresión:

$$precisión = \frac{|\{documentos\ relevantes\} \cap \{documentos\ recuperados\}}{|\{documentos\ recuperados\}}$$

De esta forma, cuanto más se acerque el valor de la precisión al valor nulo, mayor será el número de documentos recuperados que no consideren relevantes. Si por el contrario, el valor de la precisión es igual a uno, se entenderá que todos los documentos recuperados son relevantes. Esta forma de entender la precisión introduce el concepto de ruido informativo y silencio informativo. (Gomez, 2003)

## **Exhaustividad**

La exhaustividad se emplea en menor medida que la precisión. Algunos autores suelen definirlo en lengua castellana como “rellamada” procedente del término inglés “*recall*”, en otros casis como “recobrado”. Este ratio viene a expresar la proporción de documentos relevantes recuperados, comparado con el total de los documentos que son relevantes existentes en la base de datos, con total independencia de que estos, se recuperen o no. Se definió por primera vez a mediados de siglo XX. La ecuación en este caso se expresa como:

$$exhaustividad = \frac{|\{documentos\ relevantes\} \cap \{documentos\ recuperados\}}{|\{documentos\ relevantes\}}$$

Si el resultado de esta fórmula arroja como valor uno, se tendrá la exhaustividad máxima posible, y esto viene a indicar que se ha encontrado todo documento relevante que resida en la base de datos, por lo tanto no se tendrá un ruido, ni silencio informativo: siendo la recuperación de documentos entendida como perfecta. Por el contrario en el caso que el valor de la exhaustividad sea igual a cero,

se tiene que los documentos obtenidos no poseen relevancia alguna. (Gomez, 2003)

## Valor F

El valor f en estadística es la medida de precisión que tiene un test. Se emplea en la determinación de un valor único ponderado de la precisión y la exhaustividad. Se suele emplear en la fase de pruebas de algoritmos de búsqueda y recuperación de información y clasificación de documentos.

El valor F se considera como una medida armónica que combina los valores de la precisión y de la exhaustividad, de tal forma que:

$$F1 = 2 * \frac{\textit{precisión} * \textit{exhaustividad}}{\textit{precisión} + \textit{exhaustividad}}$$

La fórmula general para un número real  $\beta$  es:

$$F\beta = (1 + \beta^2) * \frac{\textit{precisión} \cdot \textit{exhaustividad}}{(\beta^2 \cdot \textit{precisión}) + \textit{exhaustividad}}$$

Si  $\beta$  es igual a uno, se está dando la misma ponderación (o importancia) a precisión que a la exhaustividad, si  $\beta$  es mayor que uno se da más importancia a exhaustividad, mientras que si es menor que uno se le da más importancia a la precisión. (Beitzel, 2006)

## Protégé

Protégé es una plataforma de código abierto que provee a una creciente comunidad de usuarios con un conjunto de herramientas para la construcción de modelos de dominio y las aplicaciones basadas en el conocimiento con ontologías. Protégé implementa un amplio conjunto de estructuras de modelado del conocimiento y las acciones que apoyan la creación, visualización y manipulación de ontologías en diversos formatos de representación. (Protégé)

## **Spring**

Spring es un *Framework* para el desarrollo de aplicaciones de código abierto y funciona bajo la plataforma Java. (Spring)

## **1.6. ORGANIZACIÓN DEL DOCUMENTO**

El presente documento de trabajo de grado se encuentra ordenado de la siguiente manera:

En el capítulo 1 se especifica el trabajo que se va a desarrollar, se define el problema y se plantea la posible solución. Se encuentra la justificación que permite entender el desarrollo de la investigación; el planteamiento de los objetivos que se van a cumplir y el marco teórico en donde se detallan cada uno de los conceptos que serán utilizados durante el progreso de los siguientes capítulos.

El capítulo 2 describe la revisión sistemática de trabajos relacionados. También se encuentran los aportes.

El capítulo 3 presenta el análisis de Metodologías de Procesamiento de Lenguaje Natural para Extracción de Casos de Uso, los tipos, los niveles y técnicas del procesamiento de texto, los problemas existentes, la descripción y comparación de las herramientas del procesamiento de lenguaje natural investigadas y la herramienta seleccionada para el desarrollo del trabajo.

En el capítulo 4 se detalla la herramienta para la asistencia en la extracción de casos de uso.

En el capítulo 5 se realiza una evaluación de la herramienta desarrollada, esto con el fin de hacer un acercamiento a la posible solución en la extracción automática de casos de uso por medio de un asistente.

Finalmente, el capítulo 6 presenta las conclusiones del proyecto de grado. En este capítulo también se especifica los trabajos futuros que se pueden desprender de este trabajo final.

**6 CAPÍTULO 2. REVISIÓN SISTEMÁTICA DE TRABAJOS RELACIONADOS**  
**6.1 DE LOS ARTÍCULOS INCLUIDOS ES POSIBLE IDENTIFICAR LOS SIGUIENTES**

N°	Nombre de artículos	1	2	3	Herramienta de procesamiento
1	IDENTIFYING DUPLICATE FUNCTIONALITY IN TEXTUAL USE CASES BY ALIGNING SEMANTIC ACTIONS.	X	X	X	OpenNLP, Mate Tools
2	TOWARD AN INTELLIGENT REQUIREMENTS TOOL FOR MODELING USE CASES AND DOMAINS IN OBJECT-ORIENTED SOFTWARE PROCESSES.	X	X		OpenNLP

**Tabla 2.** Artículos relacionados con el uso de técnicas para el procesamiento de lenguaje natural (NPL).

**6.2 CARACTERÍSTICAS**

En la tabla 2 se citan dos artículos que contienen características, las cuales crean una diferencia en los métodos usados para la generación de casos de uso.

1. La primera (1) característica hace referencia a la **base del conocimiento a partir de reglas de negocio**, en los dos artículos el uso de reglas es importante al momento de realizar estas tareas. Al modelar casos de uso y con respecto a la elección del límite del sistema; el artículo número 2 utiliza la siguiente base de conocimiento específica de dominio: A los fines del desarrollo de software, el límite del sistema generalmente se elige para ser el sistema de software (y posiblemente el hardware).

- Base de conocimiento de demostración de reglas para elegir el límite del sistema:

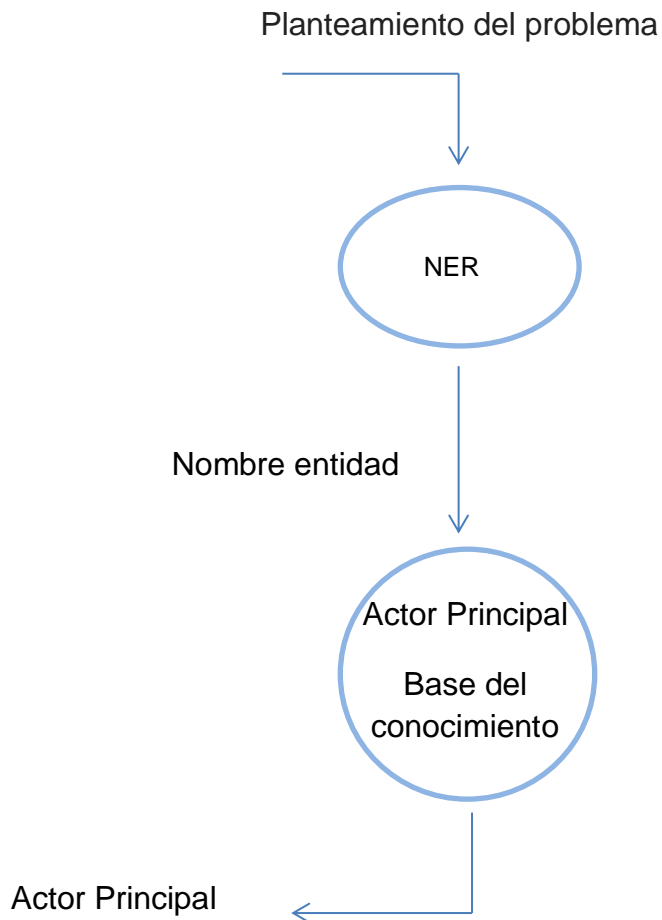
“If hay propósitos de desarrollo de software **then** generar hecho que: Limite del sistema es (software)”.

- Representación de un conocimiento de evaluación de acre:

“If hay un actor, una, y tiene objetivos de usuario que se cumplen mediante el uso de los servicios del sistema

**then** generar hecho que: Actor principal (a).”

- Flujo de datos del paso de encontrar actores primarios:



- Representación de la identificación de la base de conocimiento del usuario objetivo.

“If hay un actor, un, que tiene un objetivo, g, que cumplen con las pautas EBP  
then generar hecho que: UserGoal (a, g).”

- Representación de la base de conocimiento definitorio del usuario.

“If hay un actor, a, que tiene un objetivo de usuario, ug, then generar hecho que:  
UsaSystemTo (a, ug).”

El artículo uno (1), hace referencia a como se ayuda a los analistas a producir descripciones de requisitos de buena calidad, desarrollando un enfoque que descubre porciones textuales de casos de uso que contienen funcionalidades semánticamente similares y presenta esos escenarios como deficiencias de duplicación candidatas. El analista puede entonces intervenir y decidir si la especificación debe reformularse o enmendarse. El enfoque se basa en los avances tecnológicos recientes en NLP que hacen que el problema de la identificación de síntomas de duplicación sea tratable en términos computacionales. Inicialmente, el analista ingresa un conjunto de casos de uso de texto. Estos casos de uso son analizados por varios componentes de PNL (paso 1: Procesamiento básico del lenguaje natural) con el objetivo final de reunir conocimientos elementales sobre los escenarios textuales (es decir, cursos básicos y alternativos de los casos de uso). La primera actividad procesa los contenidos textuales sin procesar que vienen junto con las especificaciones del caso de uso. Específicamente, esta actividad ejecuta una serie de módulos NLP proporcionados por terceros y genera una serie de anotaciones para análisis posteriores (por ejemplo, detección de información duplicada).



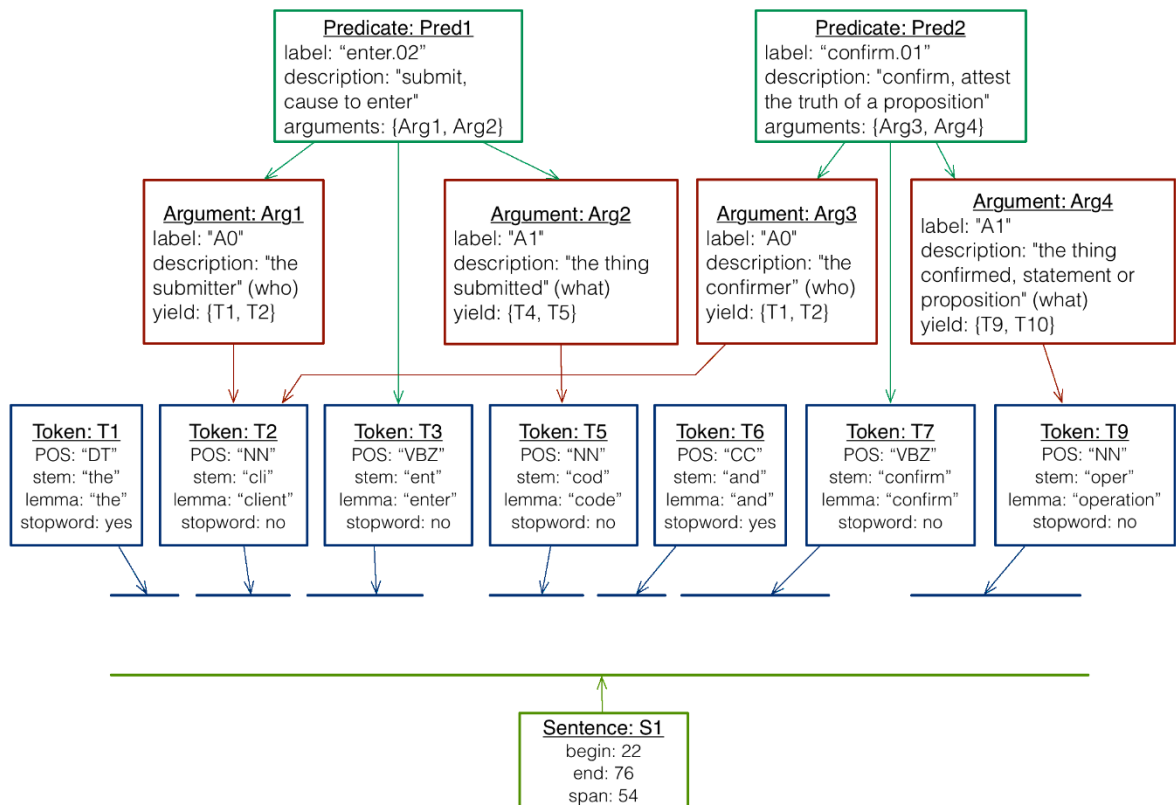


Figura 3: Labeling Domain Actions out of a Use-Case Step

2. La segunda (2) característica hace referencia a si se **ofrece una herramienta para generación de casos de uso**.

El artículo uno (1), trabaja bajo el enfoque **ReqAligner** (Analizador de requisitos con secuencia Aligner). Inicialmente, el analista ingresa un conjunto de casos de uso de texto. Estos casos de uso son analizados por varios componentes de PNL (paso 1: Procesamiento básico del lenguaje natural) con el objetivo final de reunir conocimientos elementales sobre los escenarios textuales (es decir, cursos básicos y alternativos de los casos de uso). Los componentes de PNL generan información como oraciones y límites de tokens (divisor de oraciones y tokenizador), propiedades de token (etiquetadores de parte de la voz y lematizadores) y constituyentes semánticos (predicados y argumentos con etiquetado de roles semánticos).

Una vez completados estos análisis iniciales sobre los casos de uso, se examina la intención de cada oración de las especificaciones (en el contexto de Ingeniería de Requisitos) aprovechando las particularidades del dominio de casos de uso (paso 2: Análisis Semántico de Casos de Uso con Acciones de Dominio) Esto significa que ReqAligner intenta determinar el significado de un paso de caso de uso tal como fue concebido por el analista que lo escribió.

Por ejemplo, podría predecir si un paso de caso de uso informa sobre una interacción con un sistema externo o si en realidad es una consulta para que los usuarios ingresen datos, entre otro tipo de interacciones. Con esta información, la herramienta deriva una representación abstracta de los escenarios de caso de uso en la forma de lo que llamamos acciones semánticas. Estas acciones semánticas se derivan tanto de una gran cantidad de especificaciones como de estudios recientes sobre el estilo de escritura de los casos de uso.

El artículo dos (2), se utiliza la herramienta y software **NER**, es una sub-tarea de extracción de información que busca ubicar y clasificar elementos atómicos en texto en categorías predefinidas como nombres de personas, organizaciones, ubicaciones, expresión de tiempos, valores monetarios, porcentaje, etc. Reconocimiento de entidad nombrada (NER) tiene como objetivo extraer y clasificar designadores rígidos en texto como nombres propios, especies biológicas y expresiones temporales.

3. La tercera (3) característica hace referencia al **uso de algoritmos** para la extracción de casos de uso.

Esta característica solo es tratada en el primer (1) artículo, Las contribuciones de este trabajo son dos. En primer lugar, implementan un algoritmo que permite descubrir rápidamente la funcionalidad duplicada mediante la combinación de técnicas de procesamiento de lenguaje natural (NLP) y Machine Learning (ML) de última generación con algoritmos de alineación de secuencia (SA) (normalmente utilizados en el dominio bioinformática). En segundo lugar, crean un prototipo que respalda el proceso de mejora de los casos de uso.

## 7 CAPÍTULO 3. ANÁLISIS DE METODOLOGÍAS DE PROCESAMIENTO DE LENGUAJE NATURAL PARA EXTRACCIÓN DE CASOS DE USO

### 3.1. PROCESAMIENTO DE LENGUAJE NATURAL

Un sistema de procesamiento de lenguaje natural se define como aquel que encapsula un modelo de lenguaje natural en algoritmos apropiados y eficientes. En donde las técnicas de modelado están ampliamente relacionadas con eventos en muchos otros campos, incluyendo:

Ciencia de la computación, la cual provee métodos para representar modelos, diseñar e interpretar algoritmos para herramientas de software.

Lingüística, la cual contribuye con nuevos modelos lingüísticos y procesos.

Matemáticas, la cual identifica modelos formales y métodos.

#### 3.1.1. Tipos de procesamiento de lenguaje natural

Existen dos formas de procesar el lenguaje, procesamiento por voz y procesamiento por texto:

- **Procesamiento de voz**

Es la fase de la aplicación que recibe los sonidos mediante el clásico y conocido micrófono transformándolos, luego de un largo proceso interno, en conjuntos de palabras conocidas, es decir, en una presunta frase. Esta frase puede tener o no sentido, pero es la interpretación que el módulo hace de los sonidos recibidos vía micrófono.

El procesamiento de voz se puede dividir en las siguientes categorías:

- ✓ **Reconocimiento de voz:** trata del análisis del contenido lingüístico de una señal de voz.
- ✓ **Codificación de voz:** técnicas de comprensión de señales de voz.
- ✓ **Análisis de voz:** estudio de los sonidos y la voz para propósitos de contenido lingüístico, médico, forense, entre otros.
- ✓ **Síntesis de voz:** creación de voz artificial por medio de la computadora.
- ✓ **Mejoramiento de voz:** técnicas para el mejoramiento de la inteligibilidad o la calidad en la percepción de las señales de voz.

Una vez obtenida la frase, se sigue con el análisis del lenguaje natural.

- **Procesamiento de texto**

El procesamiento de texto es más sencillo que el procesamiento de voz, ya que se procesa una frase que se encuentra escrita, solo hace recepción; evitando el procesamiento de sonidos para decodificar la misma.

Una vez obtenida la frase se procede a realizar su estudio de lenguaje. El estudio de lenguaje natural se estructura hasta en cinco niveles de análisis:

- ✓ Morfológico
- ✓ Léxico
- ✓ Sintáctico
- ✓ Semántico
- ✓ Pragmático

Se pueden incluir otros niveles de conocimiento como es el análisis del discurso, que se encarga de estudiar como la información precedente puede ser relevante para la comprensión de otra información; y, finalmente la que se denomina conocimiento del mundo, referente al conocimiento general que los habitantes tienen sobre la estructura del mundo para mantener una conversación.

(Argomedeo Pflücker & Córdor Ruiz, 2015)

### **3.1.2. Niveles del procesamiento de lenguaje natural por texto**

- **Nivel morfológico**

El nivel morfológico se encarga de analizar la composición de las palabras. El análisis de este nivel consiste en determinar la forma, la clase o categoría gramatical de cada palabra dentro de una oración. Teniendo en cuenta este nivel un sistema de procesamiento de lenguaje natural es capaz de desglosar una palabra y obtener el significado a través del significado de cada uno de sus morfemas.

- **Nivel léxico**

El nivel léxico se encarga del significado individual de cada palabra. Para poder realizar el procesamiento léxico, se debe tratar cada palabra por sí misma, y etiquetarla con una parte del discurso dependiendo del contexto en la cual se encuentra. Cada palabra que compone un texto lleva asociada un conjunto de datos morfológicos, sintácticos y semánticos. El objetivo específico de este nivel es

analizar cada una de estas palabras para saber su significado y función dentro de una oración.

- **Nivel sintáctico**

El nivel sintáctico se encarga de analizar la función de cada palabra dentro de una oración, descubriendo así la estructura u organización de la misma. El resultado de este procesamiento será una representación de la oración analizada que mostrara la relación entre las palabras que la conforman.

- **Nivel semántico**

El nivel semántico se encarga de obtener el sentido de una oración a partir de la interacción entre las palabras que la conforman. El procesamiento semántico admite solo un sentido a las palabras con varios significados, y así incluir el sentido en la representación semántica de la oración.

- **Nivel pragmático**

El nivel pragmático se encarga de analizar las diferentes variables relevantes para la comprensión de un texto o para explicar la elección de determinadas formas de llevarlo a cabo en función de los factores contextuales. Entre las variables se pueden mencionar: el emisor, el receptor, el enunciado, y en caso de tratarse de lenguaje verbal, el tono en el que se está expresando. Este nivel utiliza el contexto por encima de los contenidos del texto para la comprensión.

- **Nivel de discurso**

El nivel de discurso se encarga de trabajar con unidades de texto más grandes que los niveles anteriores. Hace foco en el texto como un todo para alcanzar el significado haciendo conexiones entre las oraciones. Dos de los procedimientos que son realizados por este nivel son la resolución de anáforas, y el reconocimiento de la estructura del texto. La resolución de anáforas consiste en realizar un reemplazo de pronombres con la entidad a la que hacen referencia.

(Ramos & Velez, 2016)

### **3.1.3. Técnicas del procesamiento de lenguaje natural por texto**

- **Detección de oraciones**

La detección de oraciones es una de las técnicas básicas correspondiente al nivel de procesamiento sintáctico. Esta técnica funciona recortando una secuencia de

caracteres entre dos signos de puntuación. El signo de puntuación debe estar acompañado por un espacio en blanco. Excluyendo el caso de la primera frase y en posibles ocasiones la última frase, para determinar las abreviaciones en el texto se utiliza palabras cargados en el modelo.

Por esto la técnica utiliza un modelo por idioma, ya que tiene símbolos o abreviaturas necesarias para detectar las sentencias.

- **Segmentación por palabras**

Una vez identificadas las oraciones que componen el texto, el siguiente paso es la segmentación por palabras, más conocida como analizador léxico o “*tokenizer*”. Esta técnica pertenece al nivel léxico y consiste en la identificación de *tokens*, las cuales son unidades lingüísticas como palabras, puntuación, números, caracteres alfanuméricos, entre otros.

- **Etiquetado gramatical o *Part Of Speech (POS)* – *Tagging***

Este proceso se encarga de asignar a cada una de las palabras de un texto su categoría gramatical de acuerdo a la definición de la misma o el contexto en el que aparece, por ejemplo, sustantivo, adjetivo, adverbio. Para ello es necesario establecer las relaciones de una palabra con sus adyacentes dentro de una frase o de un párrafo. Un mismo *token* puede obtener múltiples etiquetas POS, pero solo una es válida dependiendo del contexto.

- **Segmentación morfológica**

Un morfema es el fragmento mínimo capaz de expresar el significado de una palabra, es decir, es la unidad significativa más pequeña de un idioma.

Un morfema no es idéntico a una palabra, debido a que este puede estar acompañado, por ejemplo, por prefijos o sufijos, mientras que una palabra, por definición es independiente.

Cuando una palabra se sostiene por sí misma, se considera una raíz porque tiene un significado propio y cuando depende de otro morfema para expresar una idea, es un afixo porque tiene una función gramatical. Cada palabra puede contener uno o más morfemas.

Los morfemas se clasifican en dos categorías: Los morfemas “independientes” admiten cierta libertad fonológica del lexema y los morfemas “dependientes” van

unidos o fusionados a otra unidad mínima dotada de significado, también conocidos como monema, para completar su significado.

- **Eliminación de “*stop words*”**

Es utilizado para eliminar palabras muy comunes que pueden tener poco valor para recuperar información que necesita el usuario. La cantidad de ocurrencias de una palabra en el texto determina si es o no una “*stop word*”, dado que cuantas más ocurrencias existan menos relevancia tiene en el texto. Dentro de este grupo se encuentran los artículos, los pronombres, las preposiciones y las conjugaciones.

Esta técnica permite reducir el tamaño del texto para analizar, y se elimina aproximadamente el 30% o 40% de dichas palabras. Además, se mejora la eficiencia, ya que la selección de palabras claves es más precisa

- **Reconocimiento de entidades nombradas (NER)**

Es una sub tarea de la extracción de información que busca y clasifica elementos del texto que pertenecen a categorías predefinidas como pueden ser nombres de personas, entidades, organizaciones, lugares, expresiones temporales, cantidades, porcentajes, etc.

Para poder entender y reconocer las diferentes entidades se utilizan una serie de aproximaciones. En primer lugar, algunas entidades simples se pueden reconocer mediante patrones codificados con expresiones regulares para encontrar entidades de fecha, tiempo, velocidad, entre otros. También existen técnicas que utilizan una lista ordenada para reconocer nombres de personas, lugares, organizaciones.

Por último, existen reconocedores de entidades que utilizan un algoritmo de entropía máxima para clasificar cada uno de los *tokens* como un tipo de entidad particular en caso de que así sea.

- **Stemming**

Las palabras están morfológicamente estructuradas en prefijos, sufijos y una raíz. La técnica de *stemming* busca un concepto de la palabra eliminando tanto prefijos como sufijos y obteniendo la raíz. De esta manera, se efectúa una reducción de la palabra a su mínimo elemento con significado.

Un término que es reducido a su común denominador simplifica la recuperación de documentos cuyas palabras tenga la misma raíz.

(Ramos & Velez, 2016)



### 3.1.4. Problemas en el procesamiento de lenguaje natural

- **Variación de ambigüedad lingüística**

Cuando se habla de variación lingüística se refiere a la posibilidad de utilizar diferentes palabras o expresiones para comunicar una misma idea.

La ambigüedad lingüística se produce cuando una palabra o frase permite más de una interpretación.

Ambos fenómenos inciden en el proceso de recuperación de información, aunque de forma distinta. La variación lingüística provoca el silencio documental, es decir omisión de documentos relevantes para cubrir la necesidad de información, ya que no se han utilizado los mismos términos que aparecen en el documento.

En cambio, la ambigüedad implica ruido documental, en otras palabras, la inclusión de documentos que no son significativos, debido a que recuperan también documentos que utilizan el término, pero con significado diferente al requerido.

Estas dos características dificultan considerablemente el tratamiento automatizado del lenguaje.

A nivel léxico, una misma palabra puede adoptar diferentes roles morfo-sintácticos en función del contexto en el que aparece, ocasionando problemas de ambigüedad.

Ejemplo: “deja la comida que sobre, sobre la mesa de la cocina, dijo llevando el sobre en la mano”

La palabra “sobre” es ambigua morfológicamente ya que puede ser un sustantivo masculino singular, una preposición, y también la primera o tercera persona del presente de subjuntivo del verbo sobrar.

- **Detección de separación entre palabras**

Ocurre cuando se reciben los datos en forma oral; y esto es que en la lengua hablada no se suelen hacer pausas entre palabras y palabras.

E lugar en el que se debe separar las palabras a menudo depende de cuál es la posibilidad que mantenga un sentido lógico, tanto gramatical como contextual; lo cual tiende a realizarse utilizando técnicas probabilísticas. Esto debido a que

nuestro idioma, ya que, por ejemplo, en la lengua escrita, idiomas como el chino mandarín tampoco tiene separaciones en las palabras.

- **Complejidad del conocimiento lingüístico**

Hay dos tipos de conocimiento necesario para el reconocimiento de las estructuras en el texto y para la resolución de ambigüedades, el conocimiento general y el léxico.

El conocimiento general son los algoritmos y gramáticas, es decir, lo que aplica todo el lenguaje a cualquier oración o texto. Gramáticas detalladas y algoritmos muy sofisticados, con una profunda base matemática pueden mejorar mucho la eficacia de análisis de texto. Aunque los buenos algoritmos pueden aumentar la rapidez del proceso, en pocos casos pueden ellos mejorar la calidad de los resultados. Nunca un algoritmo astuto podrá entender el lenguaje sin un inmenso conocimiento sobre las palabras individuales.

El conocimiento léxico es específico para cada palabra, o a veces para grupos de palabras. Dependiendo de la calidad de análisis necesario, se puede requerir mucha información: cómo se conjuga la palabra, qué significa y que relaciones impide a otras palabras en la oración, qué preposiciones usa para marcar sus objetos, cuáles palabras se usan para expresar los sentidos como “muy”, “hacer”, “resultado”, entre otros. La mayoría de esa información es específica para cada idioma.

Uno de los problemas contemporáneos del procesamiento de texto es que la cantidad del conocimiento léxico necesario es tan inmensa que resulta muy difícil alcanzar el grado necesario de conocimiento en un sistema específico.

- **Complejidad del conocimiento extra lingüístico**

El problema es que el texto no comunica toda la información necesaria para entenderlo, sino que omite muchas ideas obvias, las cuales se pueden restaurar por el humano que escucha, pero no por la computadora.

- **Recepción imperfecta de datos**

Acentos extranjeros, regionalismos, o dificultades en la producción del habla, errores de mecanografiado o expresiones no gramaticales, errores en la lectura de textos mediante el reconocimiento óptico de caracteres.

(Argomedo Pflücker & Córdor Ruiz, 2015)

## **3.2. HERRAMIENTAS DE PROCESAMIENTO DE LENGUAJE NATURAL**

En la actualidad existe gran variedad de herramientas de procesamiento de lenguaje natural, a continuación, se mencionarán algunas de ellas que fueron investigadas para el desarrollo del presente trabajo.

### **3.2.1. Apache OpenNLP**

La biblioteca de Apache OpenNLP es un conjunto de herramientas basado en el aprendizaje automático para el procesamiento de texto en lenguaje natural escrito en el lenguaje de programación Java y se encuentra distribuido bajo la licencia *Apache License*. Es compatible con las técnicas más comunes de procesamiento de lenguaje natural, como detección de lenguaje, tokenización, segmentación de oraciones, reconocimiento de entidades nombradas, fragmentación, análisis sintáctico.

Las siguientes son las características de apache OpenNLP:

- Reconocimiento de entidades nombradas (NER).
- Resumen (*Summarize*): resumir párrafos, artículos, documentos, entre otros.
- Búsqueda (*Searching*): una cadena de búsqueda de búsqueda determinada puede identificarse en un texto dado, aunque la palabra dada este alterada o mal escrita.
- Etiquetado gramatical (*POS Tagging*): sirve para dividir el texto en varios elementos gramaticales para su posterior análisis.
- Traducción (*Translation*): en el procesamiento de lenguaje natural la traducción ayuda a traducir un idioma a otro.
- Agrupación de información (*Information grouping*): esta opción en el procesamiento de lenguaje natural agrupa la información textual en el contenido del documento, al igual que partes del discurso.
- Generación del lenguaje natural (*Natural language generation*): se utiliza para generar información desde una base de datos y automatizar los reportes de información, como el análisis meteorológico o informes médicos.
- Análisis de comentarios (*Feedback Analysis*): como su nombre lo indica, varios tipos de comentarios se recopilan, en relación con los productos, por parte del procesamiento de lenguaje natural para analizar qué tan exitoso es el producto.

- Reconocimiento de voz (*Speech recognition*): tiene algunas características para el reconocimiento del habla humana.

<https://opennlp.apache.org/>

### 3.2.2. NLTK

El kit de herramientas de lenguaje natural o NLTK (*Natural Language Toolkit*) es un conjunto de bibliotecas y programas diseñado para el lenguaje de programación Python, incluye demostraciones gráficas y datos de muestra. Proporciona interfaces fáciles de usar a más de 50 recursos corporales y léxicos como WordNet, junto con un conjunto de bibliotecas de procesamiento de texto para clasificación, tokenización, derivación, etiquetado, análisis y razonamiento semántico. NLTK es un proyecto de código abierto impulsado por la comunidad.

Las características que tiene NLTK son:

- Clasificación de textos
- Tokenización del texto en palabras
- Segmentación del texto en frases
- Análisis morfológico
- Etiquetado gramatical
- Tokenización y lematización
- Análisis semántico

Freeling <http://www.nltk.org/>

### 3.2.3. Freeling

Freeling es una biblioteca de C++ que proporciona servicios de análisis de lenguaje como es:

- Identificación del lenguaje
- Tokenización
- División de oraciones
- Análisis morfológico
- Detección y clasificación de entidades nombradas
- Reconocimiento de fechas, numero, magnitudes físicas
- Codificación fonética
- Etiquetado gramatical
- Análisis superficial

- Análisis de dependencia
- Etiquetado de función semántica

Freeling está diseñado para ser utilizado como una biblioteca externa desde cualquier aplicación que requiera los servicios. Si la aplicación de llamada está escrita en C++, se pueden realizar llamadas nativas a la biblioteca. Alternativamente cuenta con una interfaz de programación de aplicaciones (API) para llamar a las funciones principales de Freeling desde programas como Java, Perl, Python.

<http://nlp.lsi.upc.edu/freeling/>

### 3.3. COMPARACIÓN DE LAS HERRAMIENTAS DE PROCESAMIENTO DE LENGUAJE NATURAL

Se presenta una tabla en donde se encuentra la herramienta de procesamiento de texto con sus características y se encuentra seleccionado el idioma que soporta cada uno de ellos.

**Tabla 1.** Comparación de herramientas de procesamiento

Nombre	Lenguaje de programación	Licencia	Características	Idiomas	
				Ingles	Español
Apache OpenNLP	Java	Apache 2.0	Tokenización	x	
			Detector de oraciones	x	
			Etiquetado gramatical POS <i>Tagging</i>	x	
			Buscador de nombres	x	x
			Fragmentación	x	
			Analizador	x	
			Correferencia	x	
Freeling	C++ (APIs de Java, Python, Perl)	GPL	Tokenización	x	x
			División de oraciones	x	x
			Detección de numero	x	x
			Detección de fecha	x	x
			Diccionario morfológico	x	x
			Fijar reglas	x	x
			Detección de palabras múltiples	x	x
			Detección básica de entidad nombrada	x	x
			Detección de entidad nombrada	x	x
			Clasificación de entidad nombrada	x	x
			Detección de cantidad	x	x
			Etiquetado gramatical POS <i>Tagging</i>	x	x
			Codificación fonética	x	x
			Anotación de sentido	x	x
			Desambiguación de sentido	x	x
			Análisis superficial	x	x
Análisis completo	x	x			

			Etiquetado de rol semántico	x	x
			Resolución de correferencia	x	x
<b>NLTK</b>	Python	Apach e 2.0	Tokenización	x	
			Corpus	x	x
			Etiquetado gramatical POS <i>Tagging</i>	x	
			Morfología	x	
			Traductor	x	

## 8 CAPÍTULO 4. HERRAMIENTA PARA LA ASISTENCIA EN LA EXTRACCIÓN DE CASOS DE USO

La herramienta que se eligió es Freeling porque proporciona los servicios de procesamiento de lenguaje natural más relevantes para realizar el proyecto y admite los servicios de análisis en español que es el idioma objetivo de esta investigación.

Para realizar la validación de las características de la herramienta elegida se realiza un ejemplo de un requerimiento.



**FreeLing 4.0** - An Open-Source Suite of Language Analyzers  
*Hooked on a FreeLing?*

---

**Write your sentences**

El cliente interno requiere agregar una nueva funcionalidad el cual permita realizar la carga masiva de la base de datos del SISBÉN, esto con el fin de poder realizar los cruces entre bases de datos y generar reportes correspondientes para determinar el nivel socio económico del usuario.

**Analysis options**

- Number recognition
- Date/Time recognition
- Quantities, ratios, and percentages
- Named Entity detection
- Named Entity classification
- Multiword detection
- Phonetic encoding
- No sense annotation
- WN sense annotation: All senses
- WN sense annotation: [UKB](#) disambiguation

**Select language**: Auto-detect ▼

**Select output**: PoS Tagging ▼

Submit

Imagen 1. Captura de información a ser procesada

A continuación se muestran los resultados producto del análisis hecho por la herramienta:



## Analysis Results

### ▼ Language identification

Identified language is: Spanish (es)

### ▼ Sentences

#### Sentence 1

<b>EI</b>	<b>cliente</b>	<b>interno</b>	<b>requiere</b>	<b>agregar</b>	<b>una</b>	<b>nueva</b>	<b>funcionalidad</b>	<b>el</b>	<b>cual</b>	<b>permita</b>	<b>realiza</b>
el	cliente	interno	requerir	agregar	uno	nuevo	funcionalidad	el	cual	permitir	realizar
DA0MS0	NCCS000	AQ0MS00	VMIP3S0	VMN0000	DIOFS0	AQ0FS00	NCFS000	DA0MS0	PROCS00	VMSP3S0	VMN0000

### ▼ CONLL format

1	EI	el	DA0MS0	DA	pos=determiner type=article gen=male num=singular						
2	cliente	cliente	NCCS000	NC	pos=noun type=common gen=common num=singular						
3	interno	interno	AQ0MS00	AQ	pos=adjective type=qualificative gen=male num=singular						
4	requiere	requerir	VMIP3S0	VMI	pos=verb type=main mood=indicative tense=present person=3 num=singular						
5	agregar	agregar	VMN0000	VMN	pos=verb type=main mood=infinitive						
6	una	uno	DIOFS0	DI	pos=determiner type=indefinite gen=female num=singular						
7	nueva	nuevo	AQ0FS00	AQ	pos=adjective type=qualificative gen=female num=singular						
8	funcionalidad	funcionalidad	NCFS000	NC	pos=noun type=common gen=female num=singular						
9	el	el	DA0MS0	DA	pos=determiner type=article gen=male num=singular						
10	cual	cual	PROCS00	PR	pos=pronoun type=relative gen=common num=singular						
11	permita	permitir	VMSP3S0	VMS	pos=verb type=main mood=subjunctive tense=present person=3 num=singular						
12	realizar	realizar	VMN0000	VMN	pos=verb type=main mood=infinitive						
13	la	el	DA0FS0	DA	pos=determiner type=article gen=female num=singular						

Imagen 2. Etiquetado gramatical (POS Tagging): sirve para dividir el texto en varios elementos gramaticales para su posterior análisis.

## Analysis Results

### ▼ Language identification

Identified language is: Spanish (es)

### ▼ Sentences

#### Sentence 1

<b>EI</b>	<b>cliente</b>	<b>interno</b>	<b>requiere</b>	<b>agregar</b>	<b>una</b>	<b>nueva</b>	<b>funcionalidad</b>	<b>el</b>	<b>cual</b>	<b>permita</b>	<b>re</b>
el	cliente	interno	requerir	agregar	uno	nuevo	funcionalidad	el	cual	permitir	re
DA0MS0	NCCS000	AQ0MS00	VMIP3S0	VMN0000	DIOFS0	AQ0FS00	NCFS000	DA0MS0	PROCS00	VMSP3S0	V
1	1	0.986014	0.945545	1	0.951973	0.993421	1	1	0.930743	0.469388	1
		<b>internar</b>	<b>requerir</b>		<b>uno</b>	<b>nueva</b>			<b>cual</b>	<b>permitir</b>	
		VMIP1S0	VMM02S0		PIOFS00	NCFS000			PIOCS00	VMSP1S0	
		0.00699301	0.0544554		0.0420733	0.00657895			0.0692568	0.469388	
		<b>interno</b>			<b>1</b>					<b>permitir</b>	
		NCMS000			Z					VMM03S0	
		0.00699301			0.00586141					0.0612245	
					<b>unir</b>						
					VMM03S0						

Imagen 3. Análisis morfológico – Morphological Analysis

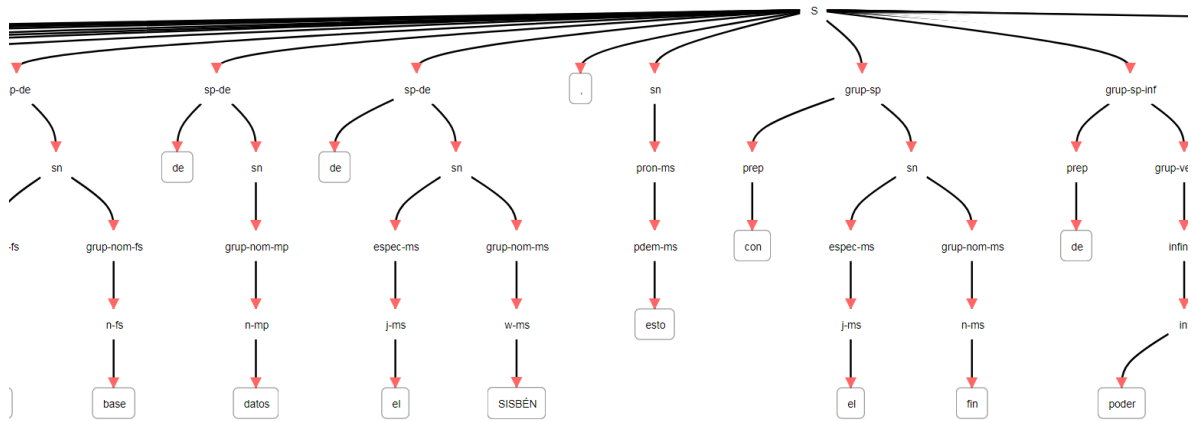


Imagen 4. Análisis superficial - Shallow Parsing

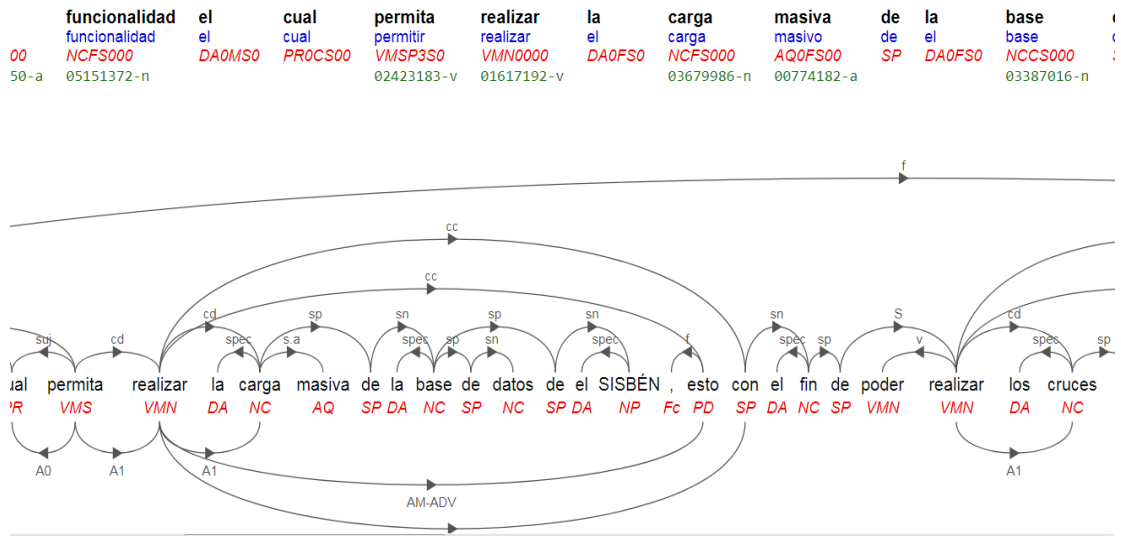


Imagen 5. Análisis de dependencia - Dependency Parsing

<http://nlp.lsi.upc.edu/freeling/demo/demo.php>

## 9 CAPÍTULO 5. EVALUACIÓN DE LA HERRAMIENTA DESARROLLADA

En este capítulo se presenta una evaluación de la herramienta Freeling, simulando la asistencia a un analista al especificar los requerimientos. Para determinar el desempeño de la herramienta, la misma fue evaluada a partir de un conjunto de escenarios de atributos de calidad obtenidos de diversas fuentes bibliográficas y sistemas reales. El objetivo principal de la evaluación fue, a partir del razonamiento mediante analogía realizado con escenarios de atributos de calidad de referencia, determinar si era posible inferir correctamente las partes de un escenario parcial. La hipótesis experimental es que la herramienta es capaz de sugerir información faltante de escenarios de atributos de calidad, permitiendo al analista que la utiliza elegir el resultado que más se ajuste al sistema que está desarrollando.

Para el análisis y comparación de la herramienta no es posible realizar la evaluación, se sugiere adoptar la evaluación propuesta en el artículo “**IDENTIFYING DUPLICATE FUNCTIONALITY IN TEXTUAL USE CASES BY ALIGNING SEMANTIC ACTIONS.**” Donde se resume las características de cada caso de estudio, se definen unas características de los casos de estudio y finalmente se hace un resumen de la solución de referencia para los casos de estudio.

## 10 CAPÍTULO 6. CONCLUSIONES Y TRABAJOS FUTUROS

En el desarrollo de un proyecto software cobran fuerza los intentos por automatizar el procesamiento de la cantidad de información que resulta de la elicitación de requerimientos, debido a que se reducen los tiempos en la elaboración de los diferentes diagramas conceptuales que son tan importantes para un buen desarrollo software. En este trabajo se abordó el problema de la extracción de casos de uso mediante procesamiento de lenguaje natural a partir de la documentación resultante de la elicitación de requerimientos.

Definir los aspectos positivos y las limitaciones que encontramos en la automatización de extracción de casos de uso mediante el procesamiento de lenguaje natural a partir de documentación preexistente.

Proponer un conjunto de pasos que permitan al ingeniero de software un enfoque en la utilización de herramientas que le permitan una mayor eficiencia en el procesamiento de información, vital para el desarrollo de un proyecto.

## 11 BIBLIOGRAFÍA

- Argomedo Pflücker, K. C., & Córdor Ruiz, A. T. (1 de Diciembre de 2015). Diseño de un sistema inteligente utilizando procesamiento de lenguaje natural (pln) para el reconocimiento de mensajes extorsivos. Trujillo, La Libertad , Perú.
- Beitzel, S. (2006). *On Understanding and Classifying Web Queries*. Chicago, Illinois, Estados Unidos .
- Ceria, S. (2001). Casos de Uso. En S. Ceria, *Cátedra Ingeniería del Software I* (pág. 1).
- Core, D. (2 de Octubre de 2017). *Dublin Core Metadata Initiative*. Obtenido de <http://www.dublincore.org/>
- Gomez, R. (2003). *La evaluación en recuperación de la información*. Obtenido de [https://www.upf.edu/hipertextnet/numero-1/evaluacion\\_ri.html](https://www.upf.edu/hipertextnet/numero-1/evaluacion_ri.html)
- Lamarca Lapuente, M. J. (8 de Diciembre de 2013). *Hipertexto*. Obtenido de Ontología: <http://www.hipertexto.info/documentos/ontologias.htm>
- Lamarca Lapuente, M. J. (8 de Diciembre de 2013). *Hipertexto*. Obtenido de Metadatos: <http://www.hipertexto.info/documentos/metadatos.htm>
- Li, K. D. (2003). Requirements capture in natural language.
- McGuinness, D., & Harmelen, F. v. (10 de Octubre de 2004). *W3C*. Obtenido de World Wide Web: <https://www.w3.org/TR/owl-features/>
- Protégé*. (s.f.). Obtenido de <https://protege.stanford.edu/>
- Ramos, F. M., & Velez, J. I. (Mayo de 2016). Integración de técnicas de procesamiento de lenguaje natural a través de servicios web. Tandil, Buenos Aires , Argentina .
- Robertson, R. y. (s.f.).
- Robertson, S. y. (2008). *Mastering the Requirements Process*. Estados Unidos.
- Salas, R. (2 de Diciembre de 2011). Diseño de un corrector ortográfico para un sistema de reconocimiento óptico de caracteres. Lima, Perú: DSpace software.
- Spring*. (s.f.). Obtenido de <https://projects.spring.io/spring-framework/>

W3C. (25 de Febrero de 2014). *RDF*. Obtenido de Web Semantic:  
<https://www.w3.org/RDF/>

W3C. (s.f.). *World Wide Web Consortium*. (W3C) Obtenido de  
<https://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>

---

<sup>i</sup> EBP (proceso comercial elemental) es un término de la ingeniería de procesos de negocios archivada, definida como: una tarea realizada por una persona en un lugar al mismo tiempo, en respuesta al evento comercial, que agrega valor comercial medible y deja los datos de manera consistente estado.