

# Aporte del Procesamiento de Lenguaje Natural a la Ingeniería de Requisitos

Claudia Litvak<sup>1,2</sup>, Jorge Doorn<sup>1,2</sup>, Facundo Velazquez Santillán<sup>1</sup>, Diego Lo Giudice<sup>1</sup>,  
Ximena Milla<sup>1</sup>, Silvina Cáceres<sup>1</sup>

<sup>1</sup> DIIT, Universidad Nacional de La Matanza, Argentina

<sup>2</sup> Escuela de Informática, Universidad Nacional del Oeste, Argentina

clitvak@unlam.edu.ar, jdoorn@exa.unicen.edu.ar,

{fvelazquezsantillan, dlogiudice, xmillla, silcaceres}@unlam.edu.ar

## Abstract

*En la Ingeniería de Software, la etapa inicial corresponde a la definición de los requisitos del sistema de software a desarrollar. Los ingenieros de requisitos generan modelos para representar el conocimiento adquirido y las decisiones tomadas en el proceso de construcción de requisitos. En particular, los modelos escritos en Lenguaje Natural son muy utilizados ya que facilitan la interacción con los clientes y usuarios. Si bien el Lenguaje Natural tiene amplios beneficios, también presenta algunas dificultades. Realizando estudios de completitud de los modelos escritos en Lenguaje Natural denominados Léxico Extendido del Lenguaje y Escenarios ha surgido la necesidad de definir estrategias específicas, tanto para crear los modelos como para realizar sobre ellos estudios de completitud. En el presente artículo se presentan resultados preliminares de mejora en la calidad de los modelos obtenidos al aplicar una herramienta de Procesamiento de Lenguaje Natural y las expectativas para trabajos futuros cuando se apliquen con mayor extensión las herramientas.*

## 1. Introducción

La Ingeniería de Requisitos es responsable por la definición de los servicios que el software deberá atender. Para atacar esta etapa inicial del desarrollo de un sistema de software, los ingenieros de requisitos desarrollan modelos, que registran el conocimiento adquirido en las diferentes etapas del proceso de definición de requisitos y las decisiones ya tomadas acerca de las características del software a construir.

Uno de los problemas que se enfrentan es la incompletitud. Ésta ha sido ampliamente estudiada en la literatura, y diferentes autores coinciden que la incompletitud es uno de los principales problemas que afronta la Ingeniería de Requisitos [1-3]. Además la

completitud ha sido considerada en variados estándares, por ejemplo el estándar IEEE 830-1998 [4] la considera una de las 8 propiedades que debe cumplir una buena especificación de requisitos de software. A su vez el estándar IEEE 29148 [5], presenta la completitud como una propiedad fundamental que debe cumplir cada requisito individualmente y el conjunto de requisitos de software como un todo.

El presente estudio se centra en los modelos desarrollados por Leite et al. [6-10], que son: el Léxico Extendido del Lenguaje (LEL) y los Escenarios. Sobre esos modelos, varios autores han atacado la completitud de los mismos [11-20].

Diferentes estudios se han efectuado a fin de evaluar la completitud de los modelos, desde los primeros trabajos realizados por Doorn y Ridaó en 2003 [11]. Posteriormente se efectuaron mejoras semánticas en las estrategias de estimación de la completitud a fin de obtener una mayor verosimilitud de los resultados. Luego se realizaron construcciones de los modelos con un mayor nivel de detalle, usando heurísticas más refinadas [15]. En todos los casos el nivel de completitud tuvo una alta coincidencia con estimaciones anteriores, donde se habían aplicado enfoques basados en aspectos sintácticos [15].

Las conclusiones a las que se arribó es este punto fueron que:

- i) Las estimaciones de completitud parecen ser confiables.
- ii) Las mejoras en las heurísticas resultaron poco relevantes, por lo que los abordajes utilizados no lograron una mejora en la eficacia en la construcción de los modelos.

A principios de 2016, Litvak et al. [21], proponen utilizar estrategias de Procesamiento de Lenguaje Natural (PNL), con el fin de mejorar el tratamiento de la completitud de los modelos.

En el presente artículo se describen las estrategias utilizadas para incorporar PNL al proceso de requisitos,

así como algunos resultados preliminares muy promisorios. Dichos estudios preliminares avizoran una importante mejora en la calidad de los modelos y han sugerido varios trabajos futuros. El núcleo de los trabajos futuros está orientado a crear heurísticas que aprovechen las herramientas de construcción de manera de mejorar la percepción del ingeniero de requisitos del problema que está siendo abordado.

## **2. Completitud de Modelos de Requisitos escritos en Lenguaje Natural**

Es apropiado insistir que para mejorar la completitud de un documento escrito en lenguaje natural es necesario disponer de mecanismos para estimarlas. Esto ocurre escasamente, ya que en la mayoría de los casos, incluyendo las normas mencionadas, todo se reduce a expresiones de deseo en las que se recomienda construir modelos lo más completos posibles, sin brindar pista alguna acerca de cómo hacerlo o cómo medirlo.

Si bien se han realizado distintas propuestas para atacar la incompletitud de los modelos LEL y Escenarios, los resultados obtenidos son aún poco satisfactorios desde el punto de vista de la completitud obtenida, pero razonablemente válidos en lo que se refiere a la medición de la misma.

Los primeros trabajos realizados por Doorn y Ridao en 2003 [11] y en 2006 [12] aplicaron el método de captura - recaptura [22] para estimar la completitud. El método de captura - recaptura fue utilizado inicialmente en otras áreas de la ciencia, como el área de la Biología, Medicina y la Ecología. Posteriormente se aplicó en el campo de la Ingeniería de Software para estimar el número total de errores presentes en un artefacto de software, a partir de los resultados de un proceso de inspección sobre dicho artefacto [23-25]. El método usado por Doorn y Ridao fue una adaptación del método de captura - recaptura, llamado Detection Profile Method [26], utilizado para predecir el número de elementos faltantes en el LEL y en escenarios, donde concluyeron que al aumentar el número de ingenieros de requisitos la cantidad de elementos estimados se acercaría significativamente a la estimación realmente existente en el dominio del problema. El trabajo se realizó sobre 9 muestras correspondientes a un mismo caso de estudio, realizadas por ingenieros de requisitos independientes.

Los estudios posteriores, mencionados en la introducción, consistieron en procurar mejorar la estimación de completitud incorporando correcciones semánticas. El primero de ellos, presentado por Litvak et al. [13], incorpora previamente correcciones semánticas y luego aplica el método estadístico Detection Profile Method [26]. Dichas correcciones semánticas corresponden al estudio de las denotaciones y connotaciones de cada símbolo del LEL, para establecer

efectivamente los sinónimos entre los símbolos existentes y si los símbolos realmente eran importantes en el contexto. En el segundo estudio se realizaron correcciones semánticas más refinadas, estudiando la relevancia, pertenencia, sinonimia y homonimia del contenido textual del modelo; luego se compararon con los resultados de los trabajos precedentes, concluyéndose que los datos disponibles anteriormente acerca de la completitud eran esencialmente correctos. Lamentablemente la completitud de cada muestra individual seguía siendo baja.

En un intento de mejorar el grado de completitud logrado hasta el momento se desarrollaron nuevos léxicos del mismo caso de estudio, con un proceso de construcción más cuidadoso. Teniendo en ese momento 12 muestras de un mismo caso de estudio, generadas independientemente, en un nuevo trabajo [15], se comprobó que la completitud individual seguía en el mejor caso en el orden del 50%.

Todos estos estudios muestran que la replicación de un modelo por parte de ingenieros de requisitos independientes:

- i) Es una técnica muy valiosa para el estudio de la completitud.
- ii) Mejora notoriamente la completitud si se reúne la información de todas las replicaciones.
- iii) Provee datos significativos para estimar apropiadamente la completitud.
- iv) Es poco viable en desarrollos prácticos; ya que duplica, triplica o eventualmente cuadruplica los costos involucrados.

Es por esto que es imperativo abordar el problema de mejorar la completitud de los modelos sin necesidad de recurrir a replicaciones. Es este el objetivo central que ha motorizado todos los estudios de completitud realizados hasta la fecha.

## **3. Aportes del Procesamiento de Lenguaje Natural a la construcción de Modelos de Requisitos**

Recién en el proyecto presentado en 2016 por Litvak et al. [21] aparece la propuesta innovadora de la incorporación de estrategias de Procesamiento de Lenguaje Natural por parte de los ingenieros de requisitos.

El Procesamiento del Lenguaje Natural (PNL o NLP en inglés) es una disciplina con una larga trayectoria. Nace en la década de 1960, como un subárea de la Inteligencia Artificial y la Lingüística, con el objeto de estudiar los problemas derivados de la generación y comprensión automática del lenguaje natural [27-28]. En los últimos años, las contribuciones que se han hecho

desde este dominio han crecido sustancialmente, permitiendo el procesamiento de ingentes cantidades de información, e incrementando su eficacia. Muestra de ello es la aplicación de estas técnicas como una componente esencial en los motores de búsqueda web, en las herramientas de traducción automática, o en la generación automática de resúmenes [29].

Las herramientas de Procesamiento de Lenguaje Natural permiten el reconocimiento de idiomas, la segmentación de oraciones, palabras y secciones, incluyendo en algunos casos un análisis morfológico, sintáctico y/o semántico [30]. Dentro del análisis morfológico se realiza la detección de palabras especiales, por ejemplo detección de nombres propios, fechas, distancias; la asignación de etiquetas para determinar para cada palabra su categoría gramatical correcta; y la desambiguación, que realiza la tarea de asignar el sentido correcto a las palabras, buscando la interpretación más adecuada en base al contexto. El análisis sintáctico permite hallar los constituyentes básicos que suelen llamarse en inglés “chunks”, como definir que el término “come” pertenece al Grupo Verbal o “pescado” al Grupo Nominal; permite además definir la estructura de la oración; establecer las funciones gramaticales y roles temáticos (como la división en sujeto, objeto y predicado). Luego el análisis semántico puede ser análisis léxico o proposicional [30].

### 3.1. Antecedentes de aplicación de PNL

Como se detalla en la sección 5, en base a las características y propiedades esperadas, se seleccionó la herramienta FreeLing [31] entre un conjunto de herramientas evaluadas.

Esta herramienta ha sido utilizada en varios proyectos industriales, como Ruby Reader, que es una aplicación para el iPhone que ayuda a los hablantes de japonés a comprender textos en inglés. Ha sido desarrollado en Tokio por la empresa CA-Mobile [32].

También se ha aplicado FreeLing en Vi-Clone, con el desarrollo de asistentes virtuales para páginas web corporativas. Algunos componentes de FreeLing se están integrando en el sistema de diálogo. Vi-Clone está financiando el desarrollo del módulo de corrección ortográfica que permitirá a FreeLing procesar frases del usuario escritas en variantes no estándar [33].

Otro producto que utiliza FreeLing es TextToSign, Traductor de texto en español al lenguaje de señas, que utiliza FreeLing para el procesamiento de texto [34].

Se emplea también en Dixio, que es un diccionario inteligente capaz de ayudar al lector de un texto, ofreciendo definiciones contextualizadas. Ha sido desarrollado por Semantix [35].

Además se aplica en Aport News, que es un portal de noticias que usa FreeLing como un preprocesador para

enriquecer texto en ruso. El resultado de la anotación se usa en la clasificación y agrupación de las noticias [36].

FreeLing se utilizó además en un software realizado para estudiar la opinión sobre los candidatos a presidente en Twitter en el año 2013 en Chile. En este caso se estudiaron dos técnicas desarrolladas para obtener el sentimiento asociado a un texto: un Método Estadístico y un Método Ontológico. Se eligió el Método Ontológico, para el que se construyen manualmente reglas, a fin de identificar el sentimiento, procesando texto libre con la librería FreeLing. Este trabajo fue desarrollado en la Universidad de Chile [37].

También la Universidad de Sonora, en diciembre de 2012, usó FreeLing para crear una herramienta que extrae conocimiento de prácticamente cualquier dominio en forma automática, principalmente de textos escritos en lenguaje natural en español [38].

Además se aplicó en la realización de la Primera aproximación para la extracción automática de Entidades Nombradas en corpus de documentos medievales castellanos, mediante FreeLing para español estándar y la variante diacrónica del español de los siglos XII al XVI [39] en un corpus formado por los documentos que componen el Libro Becerro de las Behetrías de Castilla del siglo XIV [40]. También se ha desarrollado un corpus de entrenamiento del etiquetador, y se ha usado la herramienta resultante en un estudio lingüístico sobre la evolución del uso del verbo haber [41-42].

A modo de resumen de estos trabajos mencionados anteriormente, se puede decir que todos comparten un núcleo común relacionado con la gestión del conocimiento; en algunos casos para su elicitación, en otros para su formalización y finalmente en otros para lograr un efecto pragmático.

Notoriamente gran parte de la Ingeniería de Requisitos consiste en extraer conocimiento, formalizarlo o usarlo en el diseño de los servicios que debe prestar un sistema de software. Es por eso que la inserción de estas estrategias en la Ingeniería de Requisitos ha resultado tan inmediata como eficaz.

### 3.2. Servicios necesarios de Procesamiento de Lenguaje Natural

Sólo es necesaria una mirada superficial al conjunto de herramientas de PNL para advertir la existencia de importantes diferencias en los servicios que prestan y en las propiedades de los mismos. Es así que se analizó a priori cuáles servicios son necesarios:

- Segmentar oraciones: Reconocimiento del punto al terminar una oración. Segmentación del texto en oraciones. Realiza la preparación y segmentación de la oración en el texto.
- Segmentar palabras: Segmentación del texto en palabras. Identifica las palabras en cada oración,

también se detectan números y signos de puntuación.

- Etiquetar con etiquetas POS: POS proviene de etiquetado de la oración (en inglés, “Part of speech”). Realiza el proceso de asignar a cada palabra su categoría gramatical correcta. Por ejemplo, para la frase: “El petróleo es un combustible”, El (artículo masculino singular), petróleo (sustantivo masculino singular), es (verbo presente indicativo), un (determinante), combustible (sustantivo masculino singular).
- Realizar análisis morfológico: Realiza tareas que determinan la forma, clase o categoría gramatical de cada palabra de una oración. Ejemplo de análisis morfológico: Se analiza la frase: “La niña llevaba sombrero rojo”, obteniendo: La: artículo, femenino, singular; niña: sustantivo común, femenino, singular; llevaba: verbo llevar, tercera persona del singular del pretérito imperfecto del indicativo, primera conjugación; sombrero: sustantivo común, masculino, singular; rojo: adjetivo calificativo, masculino, singular.
- Utilizar estimadores: Realizan la estimación de parámetros que establece el modelo estocástico del lenguaje a partir de un gran corpus de secuencias de palabras y de sus correspondientes secuencias de etiquetas semánticas. Un modelo estocástico de lenguaje es un modelo matemático en el cual se ha definido una función de probabilidad que calcula la probabilidad de ocurrencia de una frase en un lenguaje dado.
- Permitir la desambiguación: Realiza la tarea de asignar el sentido correcto a las palabras. Selecciona la interpretación más adecuada teniendo en cuenta el contexto. Considerando el siguiente ejemplo: “Yo **bajo** con el hombre **bajo** a tocar el **bajo bajo** la escalera”. Se presentan cuatro categorías diferentes: verbo, adjetivo, sustantivo y preposición. El desambiguador devolverá todas ellas para cada una de las apariciones de la forma “**bajo**” en la oración.

Subsidiariamente se esperaba de la herramienta:

- Ser del dominio público de manera que las herramientas que se construyan puedan ser transferidas al medio productivo sin limitaciones.
- Tener código abierto: Para facilitar la observación del funcionamiento interno de la herramienta.
- Brindar procesamiento multilingüe automático: Dado que este proyecto de requisitos tiene trabajos interrelacionados con otros países del

mundo, como Brasil especialmente, y con la posibilidad de ampliar a idioma inglés.

- Permitir utilización local: Preferentemente poder utilizar PNL sin necesidad de conexión a Internet continua, para no depender del servidor del proveedor del servicio, y por lo tanto obtener una mayor disponibilidad.
- Aplicar diccionarios accesibles y modificables: La observación de los diccionarios y la posibilidad de modificarlos facilita el funcionamiento y permite ampliar la terminología a ser aplicada.
- Facilitar la integración con otros lenguajes de programación.
- Poder ser utilizada en diversas plataformas.
- Existencia de actualizaciones mediante nuevas versiones.

#### 4. Comparación de los servicios que brindan diferentes herramientas de PNL

En la Tabla 1 se presentan un subconjunto de las herramientas más representativas analizadas. Se han evaluado en total un conjunto de 10 herramientas, presentando cada una de ellas un conjunto de servicios diferentes.

Tabla 1. Comparación de servicios de herramientas de PNL

| Función          | Segmentación de oraciones | Segmentación de palabras | Etiquetado POS | Análisis morfológico | Estimadores | Desambiguación |
|------------------|---------------------------|--------------------------|----------------|----------------------|-------------|----------------|
| FreeLing         | Sí                        | Sí                       | Sí             | Sí                   | Sí          | Sí             |
| NLTK             | Sí                        | Sí                       | Sí             | Sí                   | Sí          | Sí             |
| AS*              | Sí                        | Sí                       | Sí             | Sí                   | Sí          | Sí             |
| Grampal          | Sí                        | Sí                       | Sí             | Sí                   | No          | No             |
| WordNet          | Sí                        | Sí                       | No             | No                   | No          | Sí             |
| FrameNet español | Sí                        | Sí                       | Sí             | No                   | No          | No             |
| Snowball         | No                        | Sí                       | No             | Sí                   | Sí          | Sí             |
| Apache OpenNLP   | Sí                        | Sí                       | Sí             | No                   | No          | Sí             |
| TreeTagger       | No                        | Sí                       | Sí             | No                   | No          | Sí             |
| SVMTool          | Sí                        | Sí                       | Sí             | No                   | No          | No             |

\*AS es el Analizador sintáctico on-line

En un análisis preliminar se observó que a las herramientas Grampal [43], WordNet [44], FrameNet español [45], Snowball [46], Apache OpenNLP [47],

TreeTagger [48] y SVMTool [49] les faltaban algunos de los servicios necesarios.

De entre las tres que cumplen todas las condiciones, FreeLing [31], NLTK [50] y AS [51] se seleccionó el FreeLing por los siguientes motivos.

Respecto de AS se consideró necesario descartarla dado que no tiene código abierto ni es multi-idioma. Se plantea a futuro trabajar con este mismo software en inglés y quizás en portugués.

Si bien NLTK resulta aceptablemente eficiente para procesar texto, en idioma español se pudo observar que no presenta un buen funcionamiento para la

identificación y clasificación de entidades nombradas (nombres de organizaciones, ciudades, personas, etc.), mientras que en la herramienta FreeLing, esta funcionalidad se realiza perfectamente, tanto para el español como para otros idiomas.

A continuación se muestra un ejemplo de prueba realizado en dichas herramientas. Se ingresó la siguiente oración en ambas herramientas: “Buenos Aires es una hermosa ciudad en Argentina”.

El resultado obtenido en NLTK [50] se puede observar en la Figura 1.

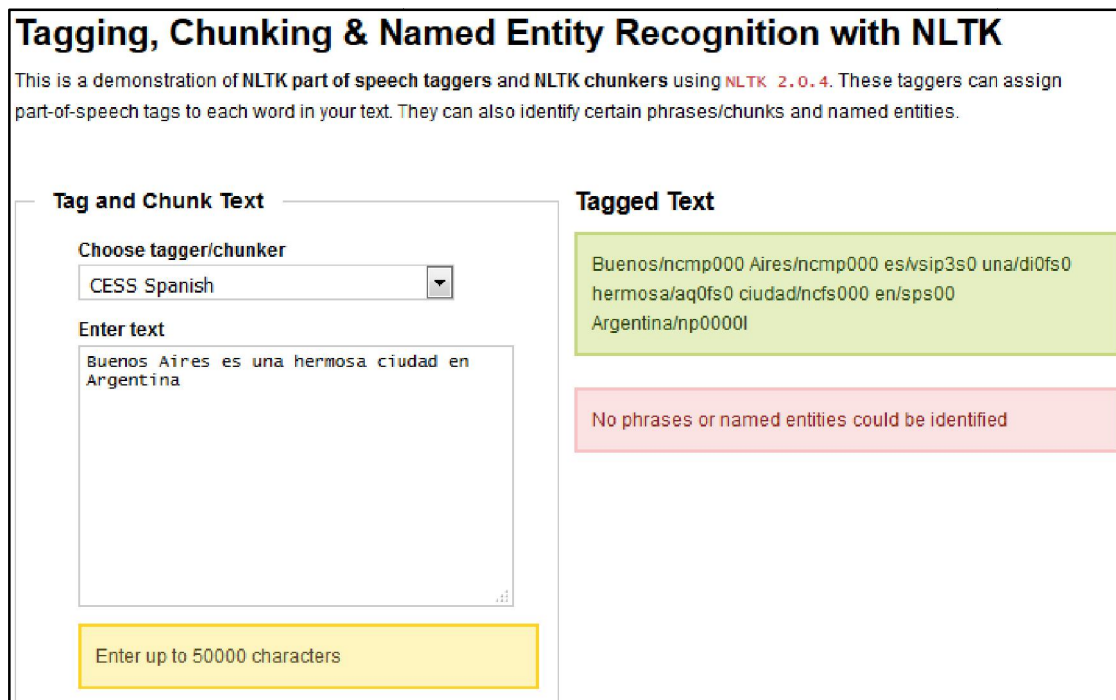


Figura 1. Ejemplo de falta de reconocimiento de entidades nombradas en NLTK

El resultado obtenido con la herramienta FreeLing [52], se presenta en la Figura 2, con la salida obtenida al ingresar la misma frase y marcar las opciones Named Entity detection y Named Entity classification (detección y clasificación de Entidades Nombradas), seleccionando auto-detección de idiomas y PoS Tagging (etiquetado PoS).

Se puede observar entonces que en la herramienta NLTK [53] la detección de entidades nombradas no se realiza de manera correcta, y por lo tanto su clasificación tampoco resulta satisfactoria.

En la frase “Buenos Aires es una hermosa ciudad en Argentina” el NLTK a “Buenos Aires” lo toma por separado como dos sustantivos comunes, siendo:

- Buenos: ncmp000, donde: n: sustantivo c: común m: masculino p: plural.

- Aires: ncmp000, donde: n: sustantivo c: común m: masculino p: plural.

En cambio FreeLing si reconoce a “Buenos Aires” como un sustantivo propio, o sea que lo reconoce como una entidad nombrada. Es decir que la herramienta FreeLing, al evaluar texto en español, detecta y clasifica las entidades nombradas de manera correcta.

Lo mismo se repite en muchos otros casos probados, tanto en FreeLing como en NLTK, con frases similares, como por ejemplo “La Sagrada Familia es la mayor representación de arte de Barcelona.”, donde NLTK no reconoce la Sagrada Familia como entidad nombrada. También en la frase “Manuel Antonio es la ciudad más bonita de Costa Rica”, donde NLTK no reconoce la ciudad de Manuel Antonio ni a Costa Rica como

entidades nombradas. Ambas entidades nombradas son perfectamente reconocidas en FreeLing.

Se procedió al estudio de las propiedades y los servicios deseados, en base a lo planteado en la Sección 3.2. Inicialmente se evaluó el origen de la herramienta FreeLing, desarrollada en TALP [54], grupo encabezado por Lluís Padró [55-56]. Se trata de un proyecto financiado parcialmente por el Gobierno Español, la Unión Europea y la empresa Vi-Clone. A principios de mayo de 2016 se ha presentado la versión 4.0 del FreeLing, manteniendo la compatibilidad y disponibilidad de las versiones anteriores. El proyecto se estructura como una librería que puede ser llamada desde cualquier aplicación de usuario que requiera servicios de análisis del lenguaje [55]. El software se distribuye como código abierto bajo una licencia GNU General Public License 3 y bajo licencia dual a empresas que deseen incluirlo en sus productos comerciales [57]. Ser de código abierto, permitió incorporar el código de otros

proyectos similares, como el módulo de desambiguación del sentido de las palabras basado en UKB [58]. Soporta las lenguas asturiano, catalán, castellano, galés, gallego, inglés, italiano, portugués, y ruso, variando los servicios de análisis de cada lengua [55].

Permite construir diccionarios en otros idiomas como se muestra en la utilización del Checo en “Usando FreeLing en Checo” [59]. Además soporta el formato de codificación de caracteres en Unicode (UTF-8), servicio útil para su posterior aplicación.

Respecto a las características y servicios específicos en el Procesamiento de Lenguaje Natural permite la segmentación de oraciones, la segmentación de palabras, etiquetado POS, el análisis morfológico, uso de estimadores, la desambiguación de texto, y el soporte multi-idioma.

| Analysis Results                     |           |            |                |               |           |                  |
|--------------------------------------|-----------|------------|----------------|---------------|-----------|------------------|
| ▼ Language identification            |           |            |                |               |           |                  |
| Identified language is: Spanish (es) |           |            |                |               |           |                  |
| ▼ Sentences                          |           |            |                |               |           |                  |
| Sentence 1                           |           |            |                |               |           |                  |
| <b>Buenos_Aires</b>                  | <b>es</b> | <b>una</b> | <b>hermosa</b> | <b>ciudad</b> | <b>en</b> | <b>Argentina</b> |
| buenos_aires                         | ser       | uno        | hermoso        | ciudad        | en        | argentina        |
| NP00G00                              | VSIP3S0   | DIOFS0     | AQ0FS00        | NCFS000       | SP        | NP00G00          |

Figura 2. Ejemplo de reconocimiento de entidades nombradas en FreeLing

## 5. Resultados preliminares de aplicación de PNL a Ingeniería de Requisitos

Se ha realizado una primera aplicación de PNL para intentar obtener símbolos del Léxico Extendido del Lenguaje a partir de documentos escritos en Lenguaje Natural. Una de las heurísticas de construcción del Léxico Extendido del Lenguaje indica clasificar los elementos obtenidos en Sujeto, Objeto, Verbo o Estado.

La herramienta ha sido planteada de modo interactivo, con el propósito de lograr dirigir la atención del ingeniero de requisitos hacia la mayor cantidad posible de aspectos potencialmente útiles del texto procesado. En otras palabras, se intenta fuertemente evitar que el constructor del LEL omita analizar semánticamente parte del documento.

La diferencia entre el uso cotidiano del vocabulario en el dominio de aplicación y los conceptos que los ingenieros de requisitos van aprendiendo sobre un macrosistema que no se conoce profundamente, puede crear gaps semánticos entre las partes involucradas (ingenieros de requisitos y clientes-usuarios). Como consecuencia de dichos gaps semánticos, es muy posible que algunos términos sean descartados en forma prematura, al ser evaluados superficialmente.

En el proceso de aplicación de PNL a Ingeniería de Requisitos, se decidió definir una herramienta que permitiera, mediante PNL, proponer posibles Sujetos, Objetos, Verbos o Estados, a través de la aplicación de FreeLing a documentos escritos en Lenguaje Natural. El mismo puede ser una entrevista escrita, un manual de usuario, un manual técnico u otro documento de relevancia del contexto.

La herramienta construida usando FreeLing incentiva un análisis más uniforme de los términos detectados por la misma, lo que redundará en una reducción de las omisiones.

Se trabajó a partir de una transcripción literal de una entrevista realizada a clientes/usuarios sobre un caso de estudio específico. Se han comparado los resultados obtenidos por ingenieros de requisitos aplicando PNL con los resultados de ingenieros de requisitos sin utilizarlo. Los resultados preliminares se muestran en la Tabla 2.

**Tabla 2. Resultados preliminares**

|                | IngR sin usar PNL | IngR usando PNL | No vistos por IngR sin usar PNL | No vistos por IngR usando PNL |
|----------------|-------------------|-----------------|---------------------------------|-------------------------------|
| <b>Sujetos</b> | 5                 | 6               | 0                               | 1                             |
| <b>Objetos</b> | 24                | 29              | 9                               | 4                             |
| <b>Verbos</b>  | 8                 | 19              | 13                              | 2                             |
| <b>Estados</b> | 9                 | 9               | 7                               | 7                             |

\*IngR es ingenieros de requisitos

En el caso de los Estados se observa una diferencia notable entre los símbolos obtenidos de una u otra manera. Por ejemplo el símbolo de tipo Estado denominado “Insumo en stock” lo infieren los ingenieros de requisitos cuando se habla de insumos y luego del stock de los mismos.

No está en oraciones cercanas, por lo que la herramienta no lo detecta. Por otro lado los ingenieros de requisitos no han visto, por ejemplo, el símbolo de tipo Estado “Producto repintado”, que si ha sido propuesto por la herramienta.

Con respecto a los verbos la aplicación de FreeLing permitió detectar más verbos.

**Tabla 3. Porcentajes obtenidos aplicando FreeLing**

|                          | Total de símbolos | Porcentaje de símbolos no vistos por los ingenieros de requisitos |
|--------------------------|-------------------|---|
| <b>Sujetos</b>           | 6                 | 17 %  |
| <b>Objetos</b>           | 33                | 27 %  |
| <b>Verbos</b>            | 21                | 62 %  |
| <b>Estados</b>           | 16                | 44 %  |
| <b>Total de símbolos</b> | 76                | 39 %  |

Los resultados son sumamente promisorios dado que la cantidad de símbolos hallados utilizando PNL es mucho mayor a la obtenida por los ingenieros de requisitos que no lo utilizaron. Los mismos se observan en la Tabla 3.

Lograr obtener un 39% de símbolos que no han sido vistos por los ingenieros de requisitos, los cuales pudieron ser sólo detectados aplicando FreeLing y los conceptos de PNL es una mejora sustancial de la calidad del modelo.

## 6. Conclusiones

Luego del trabajo realizado, y como consecuencia de los estudios preliminares, se avizoran los beneficios de aplicación de PNL en la generación de modelos de Requisitos escritos en Lenguaje Natural.

Haber aplicado FreeLing a entrevistas reales realizadas a los usuarios de un contexto dado, ha permitido obtener una amplia mejora de la completitud.

Esto es, por un lado una confirmación del grado de incompletitud de los modelos creados con las heurísticas actuales sin usar PNL, pero también es una evidencia de la herramienta que fue utilizada.

Como trabajo futuro se propone la finalización de la herramienta de generación de modelos de requisitos en lenguaje natural, aplicando FreeLing, no sólo en la búsqueda de símbolos del LEL sino en los modelos posteriores, como el modelo de Escenarios y la especificación de requisitos misma. Además se espera realizar una profundización de los estudios de completitud al aplicarla en diferentes contextos, evaluando también la completitud de los modelos posteriores del proceso.

Por otro lado se planea perfeccionar aún más las heurísticas de construcción de los modelos, de modo de generar modelos directamente más completos, con heurísticas refinadas.

La conjunción de mejores heurísticas, con el aprovechamiento de PNL, avizora grandes mejoras en la completitud de los modelos y por ende, en la calidad de los mismos.

## 7. Referencias

- [1] Kotonya G., y Sommerville I., Requirements engineering – processes and techniques, John Wiley & Sons UK, 1998.
- [2] Loucopoulos P., y Karakostas V., *System requirements engineering*, McGraw-Hill, Book company Europe, 1995.
- [3] Firesmith D.G., “Are Your Requirements Complete?”, *Journal of Object Technology*, Vol. 4, N° 1, January-February 2005, pp. 27-43, [http://www.jot.fm/issues/issue\\_2005\\_01/column3](http://www.jot.fm/issues/issue_2005_01/column3).

- [4] IEEE Computer Society, *Software Engineering Standards Committee, & IEEE-SA Standards Board, IEEE recommended practice for software requirements specifications*, Institute of Electrical and Electronics Engineers, 1998.
- [5] IEEE *Systems and software engineering - Life cycle processes - Requirements engineering*, IEEE, Nueva York, 2011.
- [6] Leite, J.C.S.P., y Oliveira, A.P.A., “A Client Oriented Requirements Baseline”, *Second IEEE International Symposium on Requirements Engineering*, IEEE Computer Society Press, 1995, pp. 108-115.
- [7] Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., y Oliveros, A., “Enhancing a Requirements Baseline with Scenarios”, *Requirements Engineering Journal*, Springer-Verlag London Ltd., Vol. 2, N° 4, 1997, pp. 184-198.
- [8] Leite J.C.S.P., Hadad G.D.S., Doorn J.H., y Kaplan G.N., “A Scenario Construction Process”, *Requirements Engineering Journal*, ISSN: 0947-3602, Vol.5, N° 1, pp. 38-61, Springer-Verlag London Ltd., Londres, Reino Unido, Julio 2000, <http://link.springer.com/article/10.1007/PL00010342>
- [9] Leite, J.C.S.P., *Gerenciando a Qualidade de Software com Base em Requisitos. Qualidade de Software: Teoria e Prática*, Prentice-Hall, Rocha A, Maldonado J, Weber K (eds), Cap. 17, 2001, pp. 238-246.
- [10] Leite, J.C.S.P., Doorn, J.H., Kaplan, G.N., Hadad, G.D.S., y Ridao, M.N., “Defining System Context using Scenarios”, en el libro “Perspectives on Software Requirements”, Kluwer Academic Publishers, EEUU, ISBN: 1-4020-7625-8, Cap. 8, 2004, pp. 169-199.
- [11] Doorn, J.H., y Ridao, M., “Compleitud de Glosarios: Un Estudio Experimental”, en *Proceedings 6th Workshop on Requirements Engineering Paracicaba*, [Universidade Metodista de Piracicaba](http://www.metodista.br/revista/revista-10-1-03-04-05-06-07-08-09-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-168-169-170-171-172-173-174-175-176-177-178-179-180-181-182-183-184-185-186-187-188-189-190-191-192-193-194-195-196-197-198-199-200-201-202-203-204-205-206-207-208-209-210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-225-226-227-228-229-230-231-232-233-234-235-236-237-238-239-240-241-242-243-244-245-246-247-248-249-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-265-266-267-268-269-270-271-272-273-274-275-276-277-278-279-280-281-282-283-284-285-286-287-288-289-290-291-292-293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-320-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-340-341-342-343-344-345-346-347-348-349-350-351-352-353-354-355-356-357-358-359-360-361-362-363-364-365-366-367-368-369-370-371-372-373-374-375-376-377-378-379-380-381-382-383-384-385-386-387-388-389-390-391-392-393-394-395-396-397-398-399-400-401-402-403-404-405-406-407-408-409-410-411-412-413-414-415-416-417-418-419-420-421-422-423-424-425-426-427-428-429-430-431-432-433-434-435-436-437-438-439-440-441-442-443-444-445-446-447-448-449-450-451-452-453-454-455-456-457-458-459-460-461-462-463-464-465-466-467-468-469-470-471-472-473-474-475-476-477-478-479-480-481-482-483-484-485-486-487-488-489-490-491-492-493-494-495-496-497-498-499-500-501-502-503-504-505-506-507-508-509-510-511-512-513-514-515-516-517-518-519-520-521-522-523-524-525-526-527-528-529-530-531-532-533-534-535-536-537-538-539-540-541-542-543-544-545-546-547-548-549-550-551-552-553-554-555-556-557-558-559-560-561-562-563-564-565-566-567-568-569-570-571-572-573-574-575-576-577-578-579-580-581-582-583-584-585-586-587-588-589-590-591-592-593-594-595-596-597-598-599-600-601-602-603-604-605-606-607-608-609-610-611-612-613-614-615-616-617-618-619-620-621-622-623-624-625-626-627-628-629-630-631-632-633-634-635-636-637-638-639-640-641-642-643-644-645-646-647-648-649-650-651-652-653-654-655-656-657-658-659-660-661-662-663-664-665-666-667-668-669-670-671-672-673-674-675-676-677-678-679-680-681-682-683-684-685-686-687-688-689-690-691-692-693-694-695-696-697-698-699-700-701-702-703-704-705-706-707-708-709-710-711-712-713-714-715-716-717-718-719-720-721-722-723-724-725-726-727-728-729-730-731-732-733-734-735-736-737-738-739-740-741-742-743-744-745-746-747-748-749-750-751-752-753-754-755-756-757-758-759-760-761-762-763-764-765-766-767-768-769-770-771-772-773-774-775-776-777-778-779-780-781-782-783-784-785-786-787-788-789-790-791-792-793-794-795-796-797-798-799-800-801-802-803-804-805-806-807-808-809-810-811-812-813-814-815-816-817-818-819-820-821-822-823-824-825-826-827-828-829-830-831-832-833-834-835-836-837-838-839-840-841-842-843-844-845-846-847-848-849-850-851-852-853-854-855-856-857-858-859-860-861-862-863-864-865-866-867-868-869-870-871-872-873-874-875-876-877-878-879-880-881-882-883-884-885-886-887-888-889-890-891-892-893-894-895-896-897-898-899-900-901-902-903-904-905-906-907-908-909-910-911-912-913-914-915-916-917-918-919-920-921-922-923-924-925-926-927-928-929-930-931-932-933-934-935-936-937-938-939-940-941-942-943-944-945-946-947-948-949-950-951-952-953-954-955-956-957-958-959-960-961-962-963-964-965-966-967-968-969-970-971-972-973-974-975-976-977-978-979-980-981-982-983-984-985-986-987-988-989-990-991-992-993-994-995-996-997-998-999-1000), Brasil, 2003, pp. 317-328.
- [12] Ridao, M.N., y Doorn, J.H., “Estimación de Compleitud en Modelos de Requisitos Basados en Lenguaje Natural”, en *Proceedings 9th Workshop on Requirements Engineering*, Rio de Janeiro, Brasil: Pontificia Universidade Católica do Rio de Janeiro, 2006, pp. 151-158.
- [13] Litvak C.S., Hadad G.D.S., y Doorn J.H., “Un abordaje al problema de completitud en requisitos de software”, en anales de *CACIC 2012 – XVIII Congreso Argentino de Ciencias de la Computación*, Universidad Nacional del Sur, Bahía Blanca, ISBN: 978-987-1648-34-4, Octubre 2012, pp. 827-836, <http://hdl.handle.net/10915/23715>.
- [14] Litvak, C.S., Hadad, G.D.S., y Doorn, J.H., “Correcciones semánticas en métodos de estimación de completitud de modelos en lenguaje natural”, en *Proceedings 16th Workshop on Requirements Engineering*, ORT University, Montevideo, Uruguay, 2013, pp. 105–117.
- [15] Litvak, C.S., Hadad, G.D.S., y Doorn, J.H., “Mejoras semánticas para estimar la completitud de modelos en lenguaje natural”, en anales de *CoNaIISI 2013, 1er Congreso Nacional en Ingeniería Informática / Sistemas de Información*, Universidad Tecnológica Nacional, Córdoba, Argentina, 2013.
- [16] Litvak C.S., Hadad G.D.S., y Doorn J.H., “Heurísticas para el modelado de requisitos escritos en lenguaje natural”, en anales de *CACIC 2014 - XX Congreso Argentino de Ciencias de la Computación*, Universidad Nacional de La Matanza, Buenos Aires, ISBN: 978-987-3806-05-6, Octubre 2014, <http://sedici.unlp.edu.ar/handle/10915/42337>.
- [17] Litvak C.S., Hadad G.D.S., y Doorn J.H., “Influencia de las nominalizaciones sobre la completitud de modelos de requisitos”, en anales de *WICC 2015 - XVII Workshop de Investigadores en Ciencias de la Computación*, anales electrónicos, Universidad Nacional de Salta, Salta, ISBN: 978-987-633-134-0, 2015.
- [18] Hadad G.D.S., Litvak C.S., y Doorn J.H., “Agregando semántica a técnicas de predicción de completitud en modelos de requisitos”, en anales de *WICC 2012 - XIV Workshop de Investigadores en Ciencias de la Computación*, anales electrónicos, Universidad Nacional de Misiones, Posadas, ISBN: 978-950-766-082-5, 2012, pp. 392-396, <http://hdl.handle.net/10915/18863>.
- [19] Hadad, G.D.S., Litvak, C.S., Doorn, J.H., y Ridao M.N., “Dealing with Completeness in Requirements Engineering”, *Encyclopedia of Information Science and Technology*, Third Edition, Editorial: IGI Global, Mehdi Khosrow-Pour (ed), Information ScienceReference, Hershey, PA, EEUU, 3ª edición, cap. 279, ISBN13: 9781466658882, Julio 2014 (10384 p.), pp. 2854-2863.
- [20] Hadad G.D.S., Litvak C.S., y Doorn J.H., “Problemas y Soluciones en la Compleitud de Modelos en Lenguaje Natural”, en anales de *CADI 2014 – II Congreso Argentino de Ingeniería*, Universidad Nacional de Tucumán, San Miguel de Tucumán, ISBN: 978-987-1662-51-7, T366, Septiembre 2014, artículo destacado mediante la aceptación para su exposición oral, además de su publicación en los anales electrónicos. <https://cadi.org.ar/index.php/trabajos-seleccionados/16-tecnologia-de-la-informacion-y-comunicacion>.
- [21] Litvak C.S., Hadad G.D.S., y Doorn J.H., “Procesamiento de Lenguaje Natural para Estudiar Compleitud de Requisitos”, en anales de *WICC 2016 – XVIII Workshop de Investigadores en Ciencias de la Computación*, anales electrónicos, Universidad Nacional de Entre Ríos, Entre Ríos, ISBN: 978-950-698-377-2, 2016.
- [22] Otis, D.L., Burnham, K.P., White, G.C., y Anderson, D.R., *Statistical inference from Capture on Closed Animal Populations*, Wildlife Monograph, 62, 1978.
- [23] Petersson, H., Thelin, T., Runeson, P., y Wohlin, C., “Capture–recapture in software inspections after 10 years research—theory, evaluation and application”, *The Journal of Systems and Software*, 72, 2003, pp. 249–264.
- [24] Kamel, A., y Sorenson, P., “The Application of CaptureRecapture Log-Linear Models To Software Inspections Data”, en *Proceedings ISESE '03 - International Symposium on Empirical Software Engineering*, 2003, pp. 213-222.

- [25] Thelin, T., "Team-based Fault Content Estimation in the Software Inspection Process", en *Proceedings ICSE '04 - 26th International Conference on Software Engineering*, 2004, pp. 263-272.
- [26] Wohlin, C., y Runeson, P., "Defect content estimations from Review Data", en *Proceedings 20th Intl Conference on Software Engineering*, Japón, 1998, pp. 400-409.
- [27] Méndez, E., y Moreiro González, J. A., "Lenguaje natural e indización automatizada", *Ciencias de la Información*, 30(3), 1999, pp. 11-24.
- [28] Vilares Ferro, J., *Aplicaciones del procesamiento del lenguaje natural en la recuperación de información en español*, Universidad da Coruña, Departamento de Computación, Tesis doctoral, Mayo de 2005, <http://coleweb.de.fi.udc.es/cole/library/ps/Vil2005a.pdf>, revisado el 4/7/16.
- [28] Vallez M., y Pedraza-Jimenez R., "El Procesamiento del Lenguaje Natural en la Recuperación de Información Textual y áreas afines", [en línea], *Hipertext.net*, núm. 5, 2007, <https://www.upf.edu/hipertextnet/numero-5/pln.html>.
- [30] Alonso i Alemany, L., "Herramientas Libres para Procesamiento del Lenguaje Natural", en *5tas Jornadas Regionales de Software Libre*, Rosario, Argentina. Noviembre de 2005, <http://www.cs.famaf.unc.edu.ar/~laura/freeNLP>, revisado el 18/5/16.
- [31] FreeLing, <http://nlp.lsi.upc.edu/FreeLing-old/>, visited on 10/5/16.
- [32] CA-Mobile, <http://www.camobile.com>, revisado el 14/5/16.
- [33] Vi-Clone, <http://www.vi-clone.com>, revisado el 14/5/16.
- [34] TextToSign, <http://www.textosign.es>, revisado el 14/6/16.
- [35] Stana I., *Using the FreeLing library with Czech Bachelor's Thesis*, Masarykova univerzita, Fakulta informatiky, Bachelor's Thesis, 2013, [http://is.muni.cz/th/256650/fi\\_b/thesis.pdf](http://is.muni.cz/th/256650/fi_b/thesis.pdf), , revisado el 28/7/16.
- [36] Aport News, <http://news.afort.ru>, revisado el 14/5/16.
- [37] Universidad de Chile- Facultad de Ciencias Físicas y Matemáticas- Departamento de Ciencias de la Computación. Presentaron el estudio Herramientas de Análisis de Opinión en Redes Sociales Virtuales en Marzo de 2014. [http://repositorio.uchile.cl/bitstream/handle/2250/116987/cf-plotouchchaj\\_i.pdf?sequence=1](http://repositorio.uchile.cl/bitstream/handle/2250/116987/cf-plotouchchaj_i.pdf?sequence=1), revisado el 14/5/16.
- [38] Universidad de Sonora, Divisiones de Ingeniería, Ciencias Exactas y Naturales y Ciencias Biológicas y de la Salud, publicado en Diciembre de 2012, <http://www.epistemus.uson.mx/revistas/pdf/numero13.pdf>, revisado el 14/5/16.
- [39] Sánchez-Marco, C., Gemma B., y Padró L., "Extending the tool, or how to annotate historical language varieties", en *Proceedings 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*.
- [40] *Janus*, Anexo 1, 2014, ISSN 2254-7290, <http://www.janusdigital.es/fa/2427888079544686815390487/hd2013baja.pdf?jsessionid=F80CB2D31139B6A7A9C4CA391738EA50>, revisado el 14/5/16.
- [41] Sánchez-Marco, C. y Stefan E., "Measuring semantic change: The case of spanish participial constructions", en *Proceedings 4th Conference on Quantitative Investigations in Theoretical Linguistics (QITL-4)*, Berlin, Germany, March 2011, pp. 79-83.
- [42] Sánchez-Marco, C., *Tracing the development of Spanish participial constructions: An empirical study of language change.*, Universitat Pompeu Fabra, Barcelona, Spain, Ph.D. tesis, 2012.
- [43] Grampal, <http://cartago.llff.uam.es/grampal/grampal.cgi>, revisado el 20/6/16.
- [44] WordNet, <http://grial.uab.es/synset/synset2.php>, revisado el 20/6/16.
- [45] FrameNet en español, <http://spanishfn.org/>, revisado el 20/6/16.
- [46] Snowball, <http://snowballstem.org/>, revisado el 20/6/16.
- [47] Apache OpenNLP, <http://opennlp.apache.org/>, revisado el 20/6/16.
- [48] TreeTagger, <http://chasen.org/~taku/software/yamcha/>, revisado el 20/6/16.
- [49] SVMTool, <http://alias-i.com/lingpipe/web/demo-ne.html>, revisado el 20/6/16.
- [50] NLTK, <http://www.nltk.org/>, revisado el 25/7/16.
- [51] Analizador sintáctico on line desarrollado por el Grupo de estructuras de datos y lingüística computacional, <http://www.gedlc.ulpgc.es>, revisado el 20/6/16.
- [52] Demo del FreeLing, <http://nlp.lsi.upc.edu/freeling/demo/demo.php>, revisado el 13/7/16.
- [53] NLTK demo, <http://text-processing.com/demo/tag/>, revisado el 25/7/16.
- [54] <http://www.talp.upc.edu/>, revisado el 1/7/16.
- [55] Padró, L., *Analizadores multilingües en FreeLing*, *Linguamatica*, 3, 2011, pp. 13-20.
- [56] Padró, L. y Stanilovsky, E., "FreeLing 3.0: Towards Wider Multilinguality", en *Proceedings Language Resources and Evaluation Conference (LREC 2012)* ELRA, Estambul, European Language Resources Association (ELRA), 2012, pp. 2473-2479.
- [57] Licencias GNU, <http://www.gnu.org/copyleft/gpl.html>, revisado el 20/5/16.
- [58] Agirre, E., y Soroa A., "Personalizing pagerank for word sense disambiguation", en *Proceedings 12th Conference of the European chapter of the Association for Computational Linguistics (EACL-2009)*, Athens, Greece, 2009, pp. 33-41.

[59] Stana I., *Using the FreeLing library with Czech Bachelor's Thesis*, Masarykova univerzita, Fakulta informatiky, Bachelor's Thesis, 2013, [http://is.muni.cz/th/256650/fi\\_b/thesis.pdf](http://is.muni.cz/th/256650/fi_b/thesis.pdf), ,  
revisado el 28/7/16.