

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



SISTEMA PARA EL RECONOCIMIENTO DE EXPRESIONES
LINGÜÍSTICAS EN ENTREVISTAS PARA INGENIERÍA
DEREQUISITOS

Reporte Técnico de Investigación presentado por:

Guadalupe Vanessa Quezada Michel 99504

Requisito para la obtención del título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Dra. Karla Miroslava Olmos Sánchez

Ciudad Juárez, Chihuahua

25 de noviembre de 2019

Ciudad Juárez, Chihuahua, a 20 de Noviembre de 2019

Asunto: Liberación de Asesoría

Mtro. Ismael Canales Valdiviezo
Jefe del Departamento de Ingeniería
Eléctrica y Computación
Presente.-

Por medio de la presente nos permitimos comunicarle que, después de haber realizado las asesorías correspondientes al reporte técnico Sistema para el reconocimiento de expresiones lingüísticas en entrevistas para ingeniería de requisitos, de la alumna Guadalupe Vanessa Quezada Michel, de la Licenciatura en Ingeniería en Sistemas Computacionales, consideramos que lo ha concluido satisfactoriamente, por lo que puede continuar con los trámites de titulación intracurricular.

Sin más por el momento, reciba un cordial saludo.

Atentamente


Karla Miroslava Olmos Sánchez
Profesor Investigador IIT



Ccp:
Coordinador del Programa de Sistemas Computacionales
Guadalupe Vanessa Quezada Michel
Archivo

Ciudad Juárez, Chihuahua, a 20 de Noviembre de 2019

Asunto: Autorización de Publicación

C. Guadalupe Vanessa Quezada Michel

Presente.-

En virtud de que cumple satisfactoriamente los requisitos solicitados, informo a usted que se autoriza la publicación del documento de Sistema para el reconocimiento de expresiones lingüísticas en entrevistas para ingeniería de requisitos, para presentar los resultados del proyecto de titulación con el propósito de obtener el título de Licenciado en Ingeniería en Sistemas Computacionales.

Sin otro particular, reciba un cordial saludo.

Mtra. Ivonne Haydee Robledo Portillo

Profesor Titular de Seminario de Titulación II

Declaración de Originalidad

Yo, Guadalupe Vanessa Quezada Michel, declaro que el material contenido en esta publicación fue elaborado con la revisión de los documentos que se mencionan en el capítulo de Bibliografía, y que la solución obtenida es original y no ha sido copiada de ninguna otra fuente, ni ha sido usada para obtener otro título o reconocimiento en otra institución de educación superior.



Guadalupe Vanessa Quezada Michel

Agradecimientos

Yo, Vanessa Quezada, quiero agradecer primeramente a Dios por permitirme llegar a la recta final de esta meta, por darme los padres que tengo, la familia bajo la que crecí, y hoy día por la familia que me permitió formar al lado de mi esposo y mis queridos hijos y por cada una de las bendiciones que me ha brindado.

Quiero agradecerle grandemente a mi asesora, la Doctora Karla Olmos, quien me dio la oportunidad de desarrollar este proyecto y quien puso la confianza en mí para poder lograrlo, por su asesoría, su apoyo y su tiempo, disposición y buen seguimiento para poder llegar a la meta propuesta.

Quiero dar gracias también, a la Profesora Ivonne Haydee Robledo, por orientarme a lo largo del desarrollo del proyecto: en la redacción del documento, en propuestas de mejora, en la retroalimentación de cada punto y lineamiento a seguir.

Dedicatoria

Yo, Vanessa Quezada, dedico este trabajo primero que a nadie a mis hijos: Kenneth y Santiago, perdón por el tiempo de descuido, de abandono quizá, pero todo fue para lograr la meta esperada, que es la culminación de mis estudios. Gracias por hacerme levantar y seguir adelante, por ser ese motor y esa motivación a lograr el objetivo, los amo con mi vida y todo el sacrificio de alguna manera fue por la búsqueda de un mejor futuro para ambos.

Dedico este trabajo también a mis padres: María de los Ángeles y Marcelo, quienes a su manera y como pudieron, me ayudaron para continuar con mis estudios, gracias por el apoyo incondicional, por hacerse cargo de mis hijos en cierto tiempo para poder asistir a la universidad, gracias a ti hermana por atender a tus sobrinos y tu ayuda para darme esta oportunidad.

Tía Mica, gracias por tu apoyo moral e incondicional, gracias por preocuparte por mis estudios, por alentarme a lograrlo, gracias por estar siempre presente.

También dedico este trabajo a mi esposo, Eric, quien de diversas maneras hizo crecer mi empeño y mi deseo por concluir la carrera, gracias.

Por último dedico muy especialmente este trabajo a dos personas que llevo en mi corazón y que ya no están con nosotros: María de la Luz y Tomas, quienes fueron un ejemplo maravilloso de personas, valores y principios.

Índice general

1. Planteamiento del Problema	3
1.1. Antecedentes	3
1.2. Definición del problema	7
1.3. Objetivo general	8
1.4. Justificación	8
1.5. Alcances y limitaciones	9
1.6. Metodología	10
2. Marco teórico	12
2.1. Conocimiento Tácito	12
2.2. Taxonomía de Bloom	14
2.3. Presuposiciones	15
2.4. Ingeniería de requisitos	18
2.5. Procesamiento de lenguaje natural	20
2.5.1. Componentes del procesamiento del lenguaje natural	20
2.5.2. Modelos para procesamiento del lenguaje natural	21

2.5.3. Alcance y técnicas de procesamiento del lenguaje natural	22
3. Desarrollo del Proyecto	25
3.1. Recolección y refinamiento de requisitos	25
3.2. Modelado, diseño rápido	26
3.3. Construcción del prototipo	28
3.4. Evaluación del prototipo por el cliente	30
3.5. Refinamiento del prototipo	31
3.6. Producto de ingeniería	33
3.7. Instalación de complementos	35
4. Resultados y discusiones	39
5. Conclusiones	46
5.1. Con respecto al objetivo de la investigación	46
5.2. Recomendaciones para futuras investigaciones	47
Bibliografía	49
A. Tabla de la Taxonomía de Bloom	52

Índice de figuras

1.1. Representación de la metodología basada en prototipos.	10
2.1. Mapa conceptual de los temas del marco teórico.	13
2.2. Tabla de niveles jerárquicos de la Taxonomía de Bloom.	16
3.1. Solución a través de código para la localización de verbos.	27
3.2. Diagrama de casos de uso general.	28
3.3. Imagen de la definición de Taxonomía de Bloom en código.	29
3.4. Código para la identificación de verbos en sus conjugaciones.	30
3.5. Pantalla de identificación de verbos.	31
3.6. Pantalla de revisión sintáctica.	32
3.7. Pantalla de autenticación de usuario.	34
3.8. Pantalla de identificación de verbos y graficación.	35
3.9. Pantalla de la lista de archivos analizados.	36
3.10. Pantalla que muestra el entorno de desarrollo Sublime Text 3.	37
3.11. Pantalla que muestra la instalación de Django.	38

4.1. Pantalla que muestra el análisis de una entrevista	40
4.2. Pantalla que muestra el reporte de verbos en una entrevista	41
4.3. Taxonomía de Bloom para ingeniería de requisitos (verbos frecuentes)	42
4.4. Gráfica con incidencia de verbos por categoría	43
4.5. Examen de usabilidad	44
4.6. Examen de usabilidad	45
A.1. Tabla de la Taxonomía de Bloom.	52

Índice de tablas

2.1. Herramientas de procesamiento de lenguaje natural.	23
---	----

Resumen

La presente investigación hace referencia al conocimiento tácito que se encuentra en las entrevistas que realiza elicitation de requisitos para el compendio de datos, dentro de la disciplina de ingeniería de requisitos, el cual se define como un problema.

Existen dos investigaciones que son la base de este proyecto, dado que ambas se concentran en el problema de la información oculta en el conocimiento tácito, el primero de ellos es “KMoS – RE: *Knowledge management on a strategy to requirements engineering*”, en el cual los autores presentan una nueva estrategia de ingeniería de requisitos para dominios de estructura informal y en los dominios de dicha estructura, no todos los conceptos y sus relaciones se definen formalmente: la mayoría de los problemas no tienen algoritmos para obtener soluciones, y los especialistas en dominios utilizan grandes cantidades de conocimiento tácito para resolver problemas, por lo que es importante utilizar una estrategia de ingeniería de requisitos adecuada para minimizarlos. La segunda investigación es la del proyecto denominado KIRM (*Knowledge Requirement Identification Model*) en el cual se propone, pero no se desarrolla la implementación de una herramienta de software que facilite la identificación de conocimiento tácito, haciendo uso de métodos y técnicas de procesamiento de lenguaje natural para la identificación de estructuras sintácticas.

Dado lo anterior, se propuso un sistema que minimice este problema y que a su vez sea una herramienta de gran utilidad para la ingeniería de requisitos, reduciendo el tiempo de análisis de entrevistas de tipo texto para poder identificar el conocimiento oculto, y tener

más clara la idea de lo que se está dando a conocer, o bien de la información a la que se quiere llegar a través de la entrevista.

Para cumplir el objetivo del proyecto propuesto, se desarrolló bajo la metodología de prototipos, la cual lo lleva hasta su culminación satisfactoriamente.

Palabras clave

Ingeniería de requisitos, Conocimiento tácito, Elicitación de requisitos, Taxonomía de Bloom, KRIM

Abstract

The present investigation refers to the tacit knowledge that is found in the interviews carried out by elicitation of requirements for the compendium of data, within the discipline of requirements engineering, which is defined as a problem.

There are two investigations that are the basis of this project, since both focus on the problem of hidden information in tacit knowledge, the first of which is “KMoS - RE: *Knowledge management on a strategy to requirements engineering*”, In which the authors present a new strategy of requirements engineering for domains of informal structure and in the domains of said structure, not all concepts and their relationships are formally defined: most problems do not have algorithms to obtain solutions, and domain specialists use large amounts of tacit knowledge to solve problems, so it is important to use an appropriate requirements engineering strategy to minimize them. The second investigation is that of the project called Krim (*Knowledge Requirement Identification Model*) in which it is proposed, but the implementation of a software tool that facilitates the identification of tacit knowledge is not developed, using methods and techniques Natural language processing for the identification of syntactic structures.

Given the above, a system was proposed that minimizes this problem and which in turn is a very useful tool for requirements engineering, reducing the time of analysis of text-type interviews in order to identify hidden knowledge, and to have clearer the idea of what is being made known, or of the information that you want to reach through the interview.

To meet the objective of the proposed project, it was developed under the prototype methodology, which brings it to completion successfully.

Keywords

Requirements Engineering, Tacit Knowledge, Requirements Elicitation, Bloom Taxonomy, KTIM

Introducción

La ingeniería de requisitos es una disciplina en la que se han propuesto técnicas, métodos y herramientas para especificar una solución a través de un sistema que cubra las necesidades del cliente, obteniendo con ello los requisitos necesarios y apropiados para el desarrollo del mismo. Este proceso requiere trabajar con los especialistas del área en la que se va a desarrollar el proyecto, ya que son quienes poseen el conocimiento. Éste, debe ser transferido a los ingenieros de requisitos, quienes lo sintetizan con su propio entendimiento y lo transforman en un documento de especificación de requisitos. El proceso se vuelve complejo debido a las grandes cantidades de conocimiento tácito e implícito, es decir, todo aquello que no es expresado por los especialistas de dominio de aplicación [1] [2].

La elicitación de requisitos implica un proceso de transferencia de conocimiento entre especialistas del área y de la solución en ingeniería de requisitos, con todas las implicaciones de interpretación que esto conlleva, como lo es la gestión del conocimiento tácito. Es decir, todo aquello que no se ha podido o sabido explicar o dar a conocer verbalmente. Este tipo de conocimiento se presenta cotidianamente y no se expresa con facilidad, ya que es difícil de identificar cuando consciente o inconscientemente se está ocultando información. Este tipo de situación se da recurrentemente en las entrevistas, cuestionarios y lluvias de ideas que forman parte de las herramientas de ingeniería de requisitos [3].

Lo que se propone en este documento es un sistema de software el cual sirva para la obtención de requisitos al facilitar la identificación de las estructuras sintácticas: como presu-

posiciones, frases que indican creencias, palabras con las cuales se pueden clasificar requisitos, y verbos de la taxonomía de Bloom. Esto con el fin de analizar si existe ambigüedad o conocimiento oculto, con el fin de tomar las medidas pertinentes para hacerlo explícito. El pasar de conocimiento tácito o implícito a explícito ayudaría a disminuir la cantidad de supuestos y (o) ambigüedades, por lo que se contaría con una mejor calidad de requisitos al evitar malas interpretaciones que pueden llevar a perder detalles relevantes en el desarrollo de la solución. Si gran parte del conocimiento permanece oculto, puede conducir al desarrollo de un sistema inadecuado, o al incremento del costo y (o) del tiempo de desarrollo del proyecto.

En el capítulo 1 se detallan los antecedentes que son la base del proyecto planteado así como también la propuesta y el alcance de cada uno de estos. En el mismo capítulo se encuentra el problema a solucionar, la justificación, el objetivo, los alcances y limitaciones y la metodología a emplear.

En el capítulo 2 se encuentran detallados todos los temas y conceptos relacionados con la literatura correspondiente al proyecto a desarrollar. Los temas a incluir son: conocimiento tácito, Taxonomía de Bloom, presuposiciones, ingeniería de requisitos y procesamiento de lenguaje natural.

Enseguida, en el capítulo 3 se detallaran los pasos que se siguieron para el desarrollo del sistema de software propuesto, mediante la metodología denominada basada en prototipos.

Posteriormente en el capítulo 4 se hará mención de los resultados obtenidos después del desarrollo del sistema de software que se ha venido mencionando. También se mostrará evidencias de dichos resultados.

Por último en el capítulo 5 se darán a conocer las conclusiones que se tienen acerca del proyecto y de sus evaluaciones, así como también se han de mencionar mejoras futuras factibles para este sistema de software.

Capítulo 1

Planteamiento del Problema

A continuación se detallaran los antecedentes que son la base del proyecto planteado, se detallará la propuesta y el alcance de cada uno de estos, para dejar ver claramente lo que ya existe, y lo que se realizará, ya que de momento, se han planteado modelos y técnicas de mejora para la disciplina de la ingeniería de requisitos pero no un sistema en particular.

1.1. Antecedentes

Este proyecto tiene como antecedente el trabajo presentado en [1] “KMoS – RE: *Knowledge management on a strategy to requirements engineering*”. En dicho documento, los autores presentan una nueva estrategia de ingeniería de requisitos para dominios de estructura informal. En los dominios de dicha estructura, no todos los conceptos y sus relaciones se definen formalmente: la mayoría de los problemas no tienen algoritmos para obtener soluciones, y los especialistas en dominios utilizan grandes cantidades de conocimiento tácito para resolver problemas. Por lo tanto, es importante utilizar una estrategia de ingeniería de requisitos adecuada para minimizarlos. El objetivo del estudio realizado es presentar la gestión del intelecto sobre una estrategia de ingeniería de requisitos (estrategia KMoS-RE): orientada a la transformación y transferencia de conocimiento, y con el objetivo de minimizar el porcentaje de información ambigua, incompleta, e información inapropiada. La funcionalidad y utilidad de

la estrategia se demostró a través de su aplicación en un caso de estudio real, dejando como muestra que el uso de KMoS-RE ayuda a internalizar el conocimiento del dominio, a aclarar la idea de la solución, a reducir la ignorancia de la simetría, a estructurar el entendimiento del dominio y a detectar y corregir creencias erróneas sobre el dominio. En este trabajo se propone utilizar el análisis del discurso como taxonomía de Bloom y presuposiciones, pero no detalla su implementación.

Posteriormente, en el 2015 en [4] ahonda en la propuesta inicial de [1] y propone un modelo denominado KRIM (*Knowledge Requirement Identification Model*) para la identificación de estructuras sintácticas que ayuden a identificar conocimiento que por alguna razón no se ha hecho explícito. Dicha propuesta es la continuación del doctorado [] y el modelo que se propone desarrollar es con la finalidad de estructurar y analizar entrevistas, así como para clasificar los diversos tipos de requisitos. Se propone hacer uso de la taxonomía de Bloom e identificación de presuposiciones para la solución de este problema. Hasta el momento, se ha encontrado que se puede hacer uso de métodos y técnicas de procesamiento de lenguaje natural para la identificación de estas estructuras sintácticas. En este trabajo que se acaba de describir, se realiza la propuesta teórica, pero no se implementa la herramienta de software que facilite la identificación de conocimiento tácito. El proyecto presentado en este documento partirá de estas dos propuestas y tendrá como finalidad materializar el modelo KRIM en un sistema de software. Para tal efecto, se dio a la tarea de investigar otros trabajos relacionados enfocados en el desarrollo de herramientas de procesamiento de lenguaje natural, los cuales se describen a continuación:

En [5] se habla de metodologías ágiles en las que exista una forma de identificar, desde el comienzo del proyecto los requisitos de software y su complejidad para hacer una estimación del tiempo requerido para el desarrollo del mismo. Estas metodologías se ven como necesarias, ya que en muchas empresas les permite reducir costos al producir productos y proporcionar servicios. SCRUM, es una de las metodologías ágiles más adoptadas por las empresas y en

esta el concepto de requisito de software se traduce como *User Stories (US)*. Una parte crucial para el éxito de un proyecto es estimar en cada iteración la complejidad y el tiempo que US requiere para su realización. Obtener esta información permite al equipo de desarrollo estimar el tiempo total que se necesitará para finalizar el proyecto. En la práctica y dado que los recursos humanos deben considerarse para cumplir con los requisitos de conocimiento, experiencia y habilidades exigidos por la US para su logro, es una actividad que todavía consume mucho tiempo y genera estimaciones pobres. Por ello se propone el uso de una nueva versión adaptada de la taxonomía de Bloom, junto con el ciclo de vida del desarrollo de software, buscando clasificar ágilmente la complejidad de los US e identificar la etapa a la que corresponde en el ciclo de vida del desarrollo de software.

El trabajo presentado en [6] describe una revisión de los sistemas de visualización o análisis de la voz y las tecnologías del habla (reconocimiento de voz, síntesis y sistemas de diálogo) que se han empleado en la enseñanza de la lengua materna o extranjera. Se abordan tanto programas comerciales como prototipos de investigación, con especial atención a los recursos para el español. Asimismo, se consideran las recomendaciones y las evaluaciones de los programas expuestas por investigadores y expertos respecto a los procedimientos de corrección, los contenidos pedagógicos o el diseño de la interfaz. Se revisaron distintas tecnologías de visualización, análisis acústico y procesamiento del habla para el aprendizaje de la lengua materna o extranjera, y los sistemas de visualización de la voz resultan ser de gran ayuda para el análisis acústico de las producciones no nativas, facilitando la investigación de aspectos fonéticos que permitan desarrollar modelos acústicos para los sistemas de procesamiento del habla. Sin embargo, requieren conocimientos de fonética acústica o el apoyo de un profesor para ser utilizadas directamente o de forma autónoma por el estudiante.

En [7], Tang desarrolló un modelo de decisión colectiva que involucra expresiones lingüísticas. Presenta una decisión colectiva modelo, que es capaz de hacer frente a evaluaciones complejas. En este modelo de decisión, las evaluaciones están representadas por expresiones

que son las fórmulas lógicas obtenidas aplicando conectivos lógicos al conjunto de etiquetas lingüísticas básicas. La presencia de vaguedad de cada expresión es implícitamente capturada por una relación de similitud semántica más que un conjunto difuso, ya que cada una determina una distribución de similitud semántica en el conjunto de etiquetas lingüísticas básicas, siendo la idea principal de este modelo colectivo de decisiones convertir las distribuciones de similitud semántica determinado por las mismas, en distribuciones de probabilidad de las expresiones correspondientes. La principal ventaja de este modelo propuesto es su capacidad para manejar complejos lingüísticos y la superposición semántica parcial entre etiquetas lingüísticas.

En el año 2013 en la Escuela Superior Politécnica del Litoral (ESPOL) se realizó la implementación de un sistema de clasificación automática para categorizar las respuestas de estudiantes a foros de discusión de acuerdo al dominio cognitivo de la taxonomía de Bloom [8]. Diversos estudios se han llevado a cabo en esta área, en esta ocasión la efectividad y desempeño de un clasificador bayesiano específicamente el clasificador Naive Bayes es analizado, empleado conjuntamente con técnicas de minería de texto y un conjunto de datos previamente clasificado por expertos. El objetivo fue determinar si el sistema de clasificación puede compararse o considerarse como un codificador humano al igual que la taxonomía de Bloom que ha sido base para muchos proyectos.

En el año 2013, bajo la reunión anual del Congreso Mundial de IFSA (*International Forestry Students Association*) y NAFIPS (*North American Fuzzy Information Processing Society*) se expuso el documento “*Eliciting Comparative Linguistic Expressions in Group Decision Making*” desarrollado por [9] de la Universidad de Jaén en España. Dicha exposición habla de la introducción de un modelo de GDM (*Group Decision Making*), que trata de las expresiones lingüísticas comparadas cercanas al lenguaje común utilizadas por expertos. Desarrollar este modelo proporciona una gramática libre de contexto para generar una comparativa de expresiones lingüísticas definidas por medio de HFLTS (*Hesitant Fuzzy Linguistic*

Terms Set).

En 1995 [10], desarrollan un trabajo que se ocupa del discurso argumentativo en escritos científicos en lengua inglesa. En dicho trabajo se revisaron los aportes de la lógica clásica, de la nueva retórica y de la inteligencia artificial, tanto para la argumentación formal y válida como para el razonamiento plausible, bien estructurado y convincente. El propósito central de este trabajo es formular una definición ampliada del concepto de argumentación que incluya la argumentación o el razonamiento de sentido común. Este razonamiento se efectúa mediante el lenguaje natural y por lo tanto es preciso conocer algunas de sus realizaciones semánticas. Como resultado del estudio, se entregan tres rejillas de análisis para construir y evaluar discursos argumentativos.

Después de revisar los trabajos mencionados, se puede observar, que aunque existen diferentes modelos planteados para la identificación de todo aquel conocimiento que por diversas razones no se da a conocer, no se ha llegado al desarrollo del sistema funcional que pueda auxiliar a los especialistas de ciertas disciplinas a enfocar de una mejor manera las ideas y necesidades. Se puede observar también, que en todos los casos, así como en el que se ha de resolver con este proyecto, el objetivo a cumplir es minimizar la cantidad de información ambigua, incompleta, e inapropiada, logrando con ello tener una transferencia de conocimiento más precisa.

1.2. Definición del problema

Actualmente la ingeniería de requisitos cuenta con un serio problema al momento de realizar el compendio de datos por medio de la entrevista a sus clientes. Éstos incluyen de alguna manera, gran cantidad de conocimiento tácito, y no dan a conocer por completo las necesidades o especificaciones del software que necesitan. Esto ocasiona que el análisis de la información abarque mas tiempo y a su vez una mala interpretación de los requisitos, y al

poner en práctica el desarrollo del sistema de software, surgen dudas o bien, al entregar el trabajo final, no se cumplen las expectativas del cliente. Por otro lado, como se mencionó en el apartado de los antecedentes, existe un modelo llamado KRIM, el cual plantea, pero no desarrolla un sistema computacional que ayude a identificar el conocimiento tácito.

1.3. Objetivo general

Identificar el conocimiento tácito a través de la Taxonomía de Bloom, con el fin de reducir el tiempo de análisis en las entrevistas de elicitación de requisitos empleando el procesamiento de lenguaje natural. Mediante un sistema de software para reconocer expresiones lingüísticas.

- Objetivos Específicos

1. Diseñar una solución de software para el reconocimiento de expresiones mediante el procesamiento de lenguaje natural.
2. Definir una base de conocimiento de expresiones lingüísticas.
3. Implementar la solución mencionada en el objetivo específico uno.
4. Evaluar el sistema en casos reales.

1.4. Justificación

El desarrollo de una solución efectiva en los proyectos de software implica cubrir las necesidades del cliente y (o) usuario. En una de las herramientas para el compendio de datos dentro de la ingeniería de requisitos como lo son las entrevistas, se da el caso de que exista conocimiento tácito, por lo que la información extraída de estas se etiqueta como incompleta. Esta dificultad, puede llegar a ocasionar que el software no cumpla con lo acordado por ambas partes. Esto puede presentarse por una transferencia inadecuada del conocimiento

entre especialistas, o en algunas ocasiones porque el cliente da por hecho; presupone que el especialista de la solución ya lo sabe, o simplemente se le olvidó mencionarlo. Ante este problema se hace necesario un sistema que ayude en la disciplina de ingeniería de requisitos a dejar al descubierto el conocimiento tácito, obteniendo a su vez un impacto tecnológico, ya que el ingeniero de software o sistemas, al tener una herramienta de apoyo para identificar el conocimiento oculto al realizar una entrevista, tendrá más clara la idea de lo que se está dando a conocer, o bien de la información a la que se quiere llegar a través de la entrevista. Dicho sistema se propone con antelación en este documento.

1.5. Alcances y limitaciones

Alcances

- El sistema a desarrollar procesará la información de tipo texto contenida en las entrevistas por medio de técnicas de procesamiento de lenguaje natural.
- El sistema a desarrollar es para la disciplina de la ingeniería de requisitos.
- Aunque se estudiará la teoría de las presuposiciones sólo se presentará una herramienta para identificación de conocimiento tácito con la utilización de la taxonomía de bloom.

Limitaciones

- Que la base a utilizar no contenga un amplio compendio de lenguaje, o bien no esté dentro del regionalismo adecuado.
- Se revisarán entrevistas textuales para el fácil manejo de la información adquirida.
- No se revisarán archivos de audio por problemas que este pueda ocasionar al momento de la detección de palabras.

1.6. Metodología

La metodología a seguir para el desarrollo del proyecto es la denominada basada en prototipos, la cual se presenta gráficamente en la Figura 1.1, para luego describirse detalladamente:

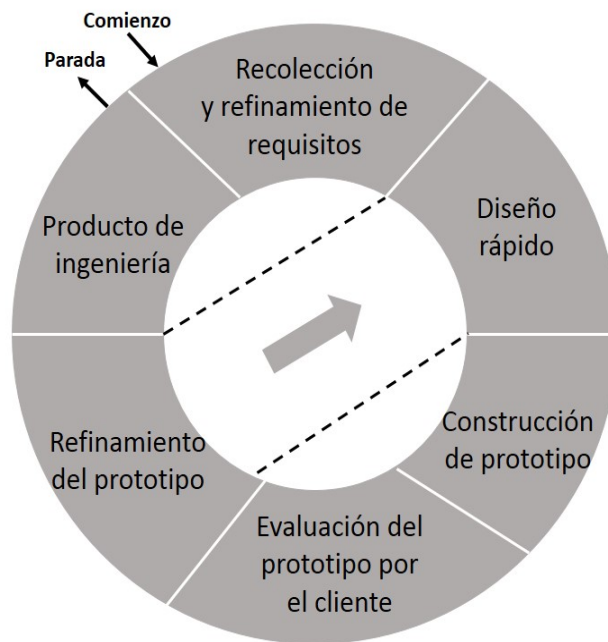


Figura 1.1: Representación de la metodología basada en prototipos.

1. Recolección y refinamiento de requisitos

En esta etapa se revisará la literatura especializada para identificar y analizar los requisitos de hardware y software necesarios para el desarrollo del sistema.

2. Modelado, diseño rápido

Se determinarán las especificaciones necesarias para el desarrollo del sistema y se diseñarán los escenarios que formaran parte del mismo, así como también el algoritmo a utilizar.

3. Construcción del prototipo

En esta etapa se realizará la codificación necesaria para la elaboración del sistema. Dicha codificación será bajo la plataforma de Python haciendo uso de las librerías NLTK y Pattern.

4. Desarrollo, evaluación del prototipo por el cliente

Para llevar a cabo la evaluación, el sistema ya debe estar terminado, así mismo las pruebas se realizarán con casos reales por el cliente.

5. Refinamiento del prototipo

En esta etapa, si el resultado de las evaluaciones anteriores es el esperado, se proseguirá con los siguientes pasos, de no ser así, se regresará a las etapas anteriores para realizar las correcciones pertinentes.

6. Producto de ingeniería

Para finalizar, se entregará al cliente el sistema con las correcciones solicitadas, y hará la ejecución del mismo para poder comprobar si resultó satisfacer sus expectativas.

Capítulo 2

Marco teórico

En este apartado se detallarán temas y conceptos relacionados con el proyecto a desarrollar, así como también una breve descripción acerca de cada uno de ellos. Los temas a abarcar son: conocimiento tácito, el cual tiene sus propias divisiones, Taxonomía de Bloom, en la cual se define una categorización de niveles de aprendizaje, presuposiciones, siendo todo aquel conocimiento previo y adquirido por una persona. Otro tema a incluir es el de Ingeniería de requisitos, siendo esta la disciplina a la que se busca ayudar por medio del proyecto planteado, auxiliado de las herramientas que forman parte de la misma. Por último se tiene el tema de procesamiento de lenguaje natural, que es una de las partes esenciales para el desarrollo del objetivo a cumplir, ya que es en donde se ofrecen herramientas que facilitan la comunicación hombre-máquina. En la Figura 2.1, se pueden ver los temas de este capítulo.

2.1. Conocimiento Tácito

El conocimiento tácito, es aquel que no se explica o que no se da a conocer. Este, es parte de la vida diaria dentro del lenguaje, la ciencia, la educación, la gestión, los deportes, el arte e incluso en la interacción con una máquina [11] [2]. En general, el conocimiento tácito comprende toda aquella información que una persona sabe y que, de alguna manera, no le es fácil expresar, ya sea consciente o inconscientemente. Es confuso e incluso inconsistente [1].

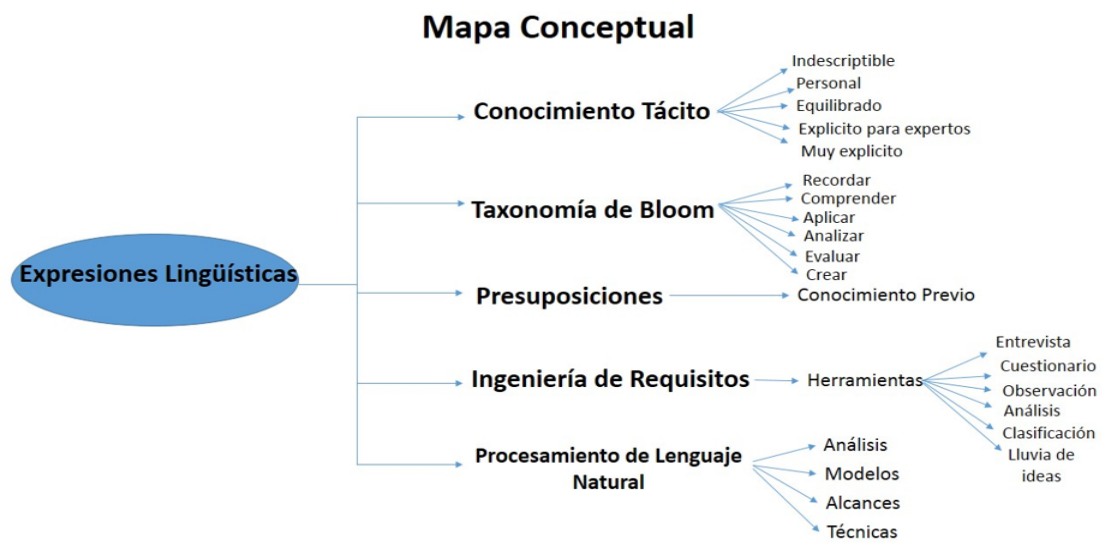


Figura 2.1: Mapa conceptual de los temas del marco teórico.

Dicho conocimiento consta de cinco divisiones que lo llevan de tácito a explícito:

Indescriptible

Comprende todo aquel conocimiento que se adquiere a lo largo de la vida, es decir tiene como base la experiencia.

Personal

Se adquiere por medio de preguntas y respuestas, haciendo uso de la entrevista y observación.

Equilibrado

Se presenta cuando la persona hace un esfuerzo para explicar y expresar el conocimiento adquirido, para así externarlo con claridad para todos.

Explícito para Expertos

Como su nombre lo indica, es para expertos, los cuales estudian el comportamiento del ser humano, como los gestos faciales o reacciones. [12].

Muy Explícita

Es cuando el conocimiento está en un nivel superior. La información se convierte en conocimiento cuando esta contiene puntos de comparación, consecuencias y contexto, es decir, depende de quien lo utiliza y de cómo lo emplea. Fundamentalmente, es a través de la interacción humana como se consigue compartir o transmitir el conocimiento tácito. Otra modalidad son las narraciones, pero deben ser guiadas por personal especializado en el área.

2.2. Taxonomía de Bloom

La Taxonomía de Bloom fue determinada por Benjamín Bloom en 1956. El objetivo de esta teoría es que el ser humano, después de realizar un proceso de aprendizaje, adquiera

nuevas habilidades y conocimientos. Define una jerarquía de niveles de aprendizaje cognitivo que van, desde el conocimiento de hechos y convenciones, hasta niveles más avanzados, que conllevan análisis, síntesis y evaluación. Dichos niveles se detallan enseguida:

- Recordar. Se refiere a todo aquello que ya no es necesario investigar, ya que es conocimiento previo que viene a la mente.
- Comprender. Trata del entendimiento de la información que se recaba acerca de un tema en específico.
- Aplicar. Se refiere a la resolución de nuevas situaciones mediante conocimiento previamente adquirido.
- Analizar. Habla de revisar a detalle la información dada para encontrar posibles causas y poder dar una solución.
- Evaluar. Es enjuiciar la información obtenida.
- Crear. Crear nuevas o alternativas soluciones para la recopilación de la información.

Para ahondar un poco más en cada uno de los niveles que conforman la taxonomía de Bloom, se presenta la siguiente Figura 2.2: Tabla de niveles jerárquicos de la Taxonomía de Bloom. [13] [14]

Para más detalles, ir a apéndice A: Taxonomía de Bloom.

2.3. Presuposiciones

La relación de una presuposición no puede ser tan explícita, a menos que se trate del significado contenido en las oraciones, explicándose desde los términos contenidos en las oraciones de cada persona, tales como: sus creencias (manera de pensar), intenciones y expectativas [15]. Existen diversos puntos de vista hacia las presuposiciones, según algunos filósofos. En

← Procesos cognitivos de orden inferior →

RECORDAR		COMPRENDER	
Recordar hechos/datos sin necesidad de entender. Se muestra material aprendido previamente mediante el recuerdo de términos, conceptos básicos y respuestas.		Mostrar entendimiento a la hora de encontrar información del texto. Se demuestra comprensión básica de hechos e ideas.	
PALABRAS CLAVE:		PALABRAS CLAVE:	
Elegir	observar	Preguntar	esquemmatizar
Copiar	omitir	Generalizar	predecir
Definir	rastrear	Clasificar	dar ejemplos
Decir	cuándo	Comparar	relacionar
Citar	repetir	Contrastar	ilustrar
Leer	relacionar	Parafrasear	demonstrar
Quién	listar	Informar	discutir
Recitar	escribir	Inferir	revisar
Cómo	dónde	Interpretar	mostrar
Por qué	reconocer	Explicar	resumir
		Expresar	observar
		Traducir	
ACCIONES	RESULTADO	ACCIONES	RESULTADO
Describir	Definición	Clasificar	Colección
Encontrar	Hechos	Comparar	Ejemplos
Identificar	Etiquetado	Ejemplificar	Explicación
Listar	Listado	Explicar	Etiquetado
Localizar	Cuestionario	Inferir	Listado
Nombrar	Reproducción	Interpretar	Esquema
Reconocer	Test	Parafrasear	Cuestionario
Recuperar	Cuaderno	Resumir	Resumen
	Fotocopia		Muestra y cuenta

Figura 2.2: Tabla de niveles jerárquicos de la Taxonomía de Bloom.

lo que respecta al estudio de la lingüística, comprende tres aspectos [16]:

1. Hablar un determinado idioma comprende ciertos actos de habla, y todo aquello que conforma dicho idioma puede simular un comportamiento lingüístico, convenientemente basado en reglas constitutivas del mismo.

2. La aseveración de la teoría de que el entranamiento y las presuposiciones lógica y semántica son diferentes, ya que su grado de implicación difiere y para determinarlo se basa en las siguientes fórmulas:

a. Entranamiento: A entranía semánticamente B si, y sólo si, en cada situación, A es verdadero, B es verdadero.

b. Presuposición lógica: A presupone una declaración B, si y sólo si:

i. A es verdadera, entonces B es verdadera.

ii. A es falsa, entonces B es verdadera.

c. Presuposición semántica: la oración A presupone semánticamente la oración B, sólo si:

i) $A _ B$

ii) $\neg A _ B$

3. Tras la investigación de varios filósofos, concuerdan en la conceptualización del enunciado y de la presuposición, pero difieren en cuanto a los indicativos que expresan la cancelación de una presuposición.

El objetivo de estudiar las presuposiciones es hacerla saber y destacarla como una propiedad dentro de las lenguas naturales del ser humano, y para identificarlas dentro de los actos de habla se han clasificado de la siguiente manera:

- Las presuposición que tienen como significado las creencias del ser humano.
- Las presuposiciones que tienen como antecedente entidades ya con significado (denota-

das).

- Las presuposiciones que se conforman por palabras o expresiones propias de la lengua natural del ser humano.

Las presuposiciones básicas de una oración simple se pueden determinar entonces a partir de las coincidencias léxicas y de la forma o contexto de la frase [17].

2.4. Ingeniería de requisitos

La ingeniería de requisitos es una disciplina que se encuentra activa dentro de la producción o desarrollo de software. Esta disciplina es de gran importancia, ya que se encarga de recopilar todas y cada una de las especificaciones que debe tener un sistema. Las especificaciones son las que permiten dar a conocer a los ingenieros de requisitos las necesidades o deseos del modelo final, solicitado por el cliente [[17]. Los ingenieros de requisitos se apoyan en ciertas técnicas y herramientas que les permiten conocer o llegar con más certeza a las especificaciones realmente necesarias para el desarrollo del software.

La ingeniería de requisitos, a través de estas técnicas y herramientas, permiten desarrollar un proyecto en forma organizada, con pasos bien estructurados siguiendo un cronograma de actividades, y puede disminuir costos, ya sea por retraso, o bien por la corrección de errores que no se detectaron con antelación a de la entrega del software. Si se logra hacer un buen uso de las técnicas y herramientas en la elicitación de requisitos, se mejorará la calidad del software a desarrollar, y con ello se evita el rechazo o el incumplimiento con el cliente. A continuación se dan a conocer algunas de las herramientas que se utilizan con mayor frecuencia en la ingeniería de requisitos [18].:

Entrevista

Es la técnica más popular para el compendio de información dentro de la ingeniería de

requisitos. La entrevista facilita la obtención de la información de manera verbal, principalmente, a través de una serie de preguntas que van del especialista hacia el cliente. Normalmente en una entrevista se va de lo más simple a lo específico, por ello se clasifican en dos tipos, las no estructuradas que es con las que se da inicio, y las estructuradas para finalizar.

Cuestionario

Es prácticamente una entrevista, pero a diferencia de la anterior, está escrita en papel u otro medio. En los cuestionarios, el cliente tiene la libertad de tomarse más tiempo para responder, en un momento y lugar adecuado. El objetivo de los cuestionarios es proporcionar información más a detalle, ya que en la entrevista verbal pueden existir preguntas inapropiadas por parte del especialista, o bien información mal detallada por parte del cliente.

Observación de tareas habituales

Esta técnica consiste en observar el proceso de trabajo de las personas, pero sin interrumpirles. El propósito de esta técnica es acercarse a la solución del problema a través del análisis realizado.

Análisis de protocolo (AP)

Esta técnica se asemeja a la anterior, pero lo que las hace diferentes es que en esta el cliente si interrumpe de alguna manera el proceso, ya que debe listar en voz alta y en orden los pasos a seguir para la realización de una tarea. Esta información es plasmada en un protocolo para analizarla y posteriormente convertirla en razonamientos que determinan una solución.

Clasificación de conceptos o agrupación de tarjetas

Esta técnica muestra la organización mental de la información que un individuo posee. También permite conocer la jerarquía en la que la almacena y que información determina como prioritaria. El proceso para realizar esta técnica es simple, ya que el individuo solo debe de ordenar un grupo de tarjetas, basándose en características que son definidas por él mismo.

Lluvia de ideas

Esta técnica consiste en reunir a un grupo de personas quienes aportaran sus ideas acerca de algún tema en específico. Una vez recolectadas, todas las ideas se analizan, para llegar a una conclusión, o bien, a la obtención de una nueva y más completa idea.

2.5. Procesamiento de lenguaje natural

El procesamiento del lenguaje natural (PLN), es una subdisciplina de la inteligencia artificial, que se especializa en emplear herramientas de la computación para una clara comunicación hombre-máquina de los lenguajes naturales [19].

A través de la comunicación hombre-máquina por medio de PLN, la computadora debe ser capaz de entender las indicaciones que se le están proporcionando. El emplear dichos lenguajes naturales ayuda al desarrollo de sistemas de software que se relacionan con el lenguaje mismo, o bien, facilita la comprensión de aquellos modelos computacionales que tienen como finalidad mostrar los mecanismos del ser humano ligados con el lenguaje [20].

Los lenguajes del ser humano pueden expresarse por escrito, oralmente, y/o mediante signos [21]. Hoy en día, el procesamiento de lenguaje natural cuenta con más avances respecto a las expresiones por escrito, ya que estas contienen más datos y son más fáciles de convertir a formato digital. Por su parte, los audios aun cuando se encuentren en formato digital, se deben procesar para llegar a tenerlos en un escrito.

2.5.1. Componentes del procesamiento del lenguaje natural

A continuación, se mencionan algunos componentes que forman parte del procesamiento del lenguaje natural. Cabe mencionar que no todos los componentes que se listarán tienen que estar presentes en una sola tarea, esto dependerá del problema a resolver:

1. Análisis morfológico o léxico. Se refiere al análisis de cada palabra para extraer unidades léxicas específicas.

2. Análisis sintáctico. Comprende el análisis de la estructura de las oraciones de acuerdo con el modelo gramatical que se utilizó.

3. Análisis semántico. Se encarga de dar una interpretación de las oraciones, ya sin ambigüedades.

4. Análisis pragmático. Realiza un análisis del contexto para dar una interpretación final.

2.5.2. Modelos para procesamiento del lenguaje natural

La computación se encarga de implementar el modelo lingüístico para los ingenieros informáticos mediante un código eficiente y funcional. Este tipo de implementación consta de dos diferentes modelos [22]:

Modelo lógico gramatical

Se refiere a aquellas reglas gramaticales que deben acompañar a cada palabra, estas reglas junto con la información adquirida en diccionarios digitales, definen las posibles soluciones para la tarea. Este tipo de modelo busca reflejar la estructura lógica del lenguaje.

Modelos probabilísticos del lenguaje natural basados en datos

Este tipo de modelos calcula la frecuencia y la probabilidad de que diferentes letras, palabras y oraciones se presenten en determinado contexto. Una vez calculada esta probabilidad, se puede asumir cuál será la siguiente unidad lingüística y en qué contexto se dará.

2.5.3. Alcance y técnicas de procesamiento del lenguaje natural

El procesamiento de lenguaje natural por lo ya anteriormente mencionado es empleado en diversas disciplinas para resolver diferentes tipos de problemas y algunas de las áreas de aplicación que abarca son [23]:

- Búsqueda
- Traducción
- Operaciones lógicas
- Agrupación de información
- Partes de etiquetado de voz
- Análisis sentimental
- Responder preguntas
- Reconocimiento de voz
- Elaboración de lenguaje natural

PLN utiliza diversas técnicas para resolver los diferentes tipos de problemas que se puedan presentar en las áreas mencionadas, ya que verifica que el modelo a seguir para dar una solución sea el correcto y poder aplicarlo.

Uno de los procesos empleado por PLN para realizar un análisis es nombrado como tokenization, este indica que el texto a de dividirse en una secuencia de palabras para trabajar con las mismas por separado.

Otro de los procesos a utilizar es nombrado como lematization, y con este se define la forma inicial de una palabra, nombrando así a esta parte del proceso como lema.

Por último, existe otro proceso llamado POS Tagging, el cual se encarga de asignarle a cada palabra una etiqueta gramatical.

Herramienta	Modelos Pre-Entrenados	Modelos en Lenguaje Español
Apache Open NLP	*Tokenizer *Sentence Detector *POS Tagger *Name Finder *Chunker *Parser	*Name Finder
Stanford Core NLP	*Tokenizer *Sentence split *POS Tagger *Lemma *Name Entities	*Token *POS Tagger
NLTK Python	*Tokenizer *Lemmatization *Stemming *POS Tagging *Chunker *Name Entities	*Tokenizer *POS Tagging Stemming
Freeling	*Tokenizer *Lemmatization *POS Tagging *Dependency Parsing Rule File	*Tokenizer *Lemmatization *POS Tagging

Tabla 2.1: Herramientas de procesamiento de lenguaje natural.

Para el desarrollo del sistema que se plantea en este documento, y en el cual se propone realice la identificación de aquellas estructuras sintácticas que muestren conocimiento tácito, se propone hacer uso de una herramienta de procesamiento de lenguaje natural y para ello es que se requiere tokenizar, POS Tagging, y la lematization.

Después de analizar diversas herramientas de procesamiento de lenguaje natural, se concluyó que la mejor opción para trabajar es NLTK, ya que es la que incluye los procesos anteriormente mencionados. Existen más herramientas, pero se encuentran en otros idiomas.

Para dejar más en claro la parte de las herramientas analizadas se muestra la Tabla 2.1: Herramientas de procesamiento de lenguaje natural, en la cual se mencionan los modelos que comprende cada una:

Es así como, haciendo uso de las librerías de lenguaje natural de NLTK y Pattern, el sistema de reconocimiento de expresiones lingüísticas se apoyará en la plataforma de Phyton.

Capítulo 3

Desarrollo del Proyecto

En este capítulo se detallaran los pasos que se siguieron en la metodología empleada, que fue la denominada basada en prototipos. Se hará mencionando todo lo que se utilizó para el desarrollo del sistema, tanto software como hardware, algoritmo, escenarios, diagramas de casos de uso y más. También se mostrará código fuente para visualizar los algoritmos empleados y la estructura del mismo.

3.1. Recolección y refinamiento de requisitos

En esta etapa se revisó la literatura especializada para identificar y analizar los requisitos de hardware y software necesario para el desarrollo del sistema de software. En hardware se hizo uso de una computadora laptop ASUS de 4gb de memoria, con sistema operativo de Windows 7, de 64 bits. En lo que respecta a software se utilizó la plataforma de Python, docx2txt para que lean los archivos de tipo texto, también se utilizaron la librería NLTK y Pattern para el manejo del lenguaje natural, así mismo, para que el sistema se encuentre en la web se hizo uso del framework Django, que aparte de ser gratuito es de código abierto escrito en Python.

3.2. Modelado, diseño rápido

Como especificaciones necesarias para el desarrollo del sistema se encontraron las siguientes:

1. Trabajar sobre la plataforma de Python. Se trabajó en la plataforma de Python, para este caso con la versión 3.5.2. Se eligió por ser un lenguaje de código abierto y un poco más práctico.

2. Trabajar con la librería NLTK y Pattern para el procesamiento de lenguaje natural. Se visualizó a estas dos librerías como las más indicadas después de analizar diversas herramientas de procesamiento de lenguaje natural. Dicho análisis se muestra en la Tabla 2.1. Respecto a la librería NLTK, se utilizó la versión 3.4.4 y de Pattern la versión 3.6.

3. Identificar verbos de la Taxonomía de Bloom. Este punto es la parte fundamental del proyecto, ya que la identificación de verbos se realiza en sus n conjugaciones. Los detalles se muestran en el inciso b, ya que es donde se muestra la solución que hizo posible lo antes mencionado.

Como escenarios para este sistema de software se tiene solo uno, que es la pantalla de la cual dependerá el mismo y en la cual se puede hacer la revisión del archivo de texto seleccionado. Dicha pantalla se describe más adelante en refinamiento del prototipo, nombrada como Figura 3.6.

En esta etapa se generó la solución a través de código para el desarrollo del sistema de software, dicha solución está basada en funciones que hacen posible la localización de los verbos de la Taxonomía de Bloom en su n conjugaciones, tal como se muestra en la Figura 3.1:

Para diseñar y desarrollar de una mejor manera el sistema planteado, también se realizó un diagrama de casos de uso. Para este caso solo se requirió de uno general, que es para

```
3 from nltk import word_tokenize  ##Importamos solo la funcion word_tokenize de la libreria
4 from nltk.tokenize.treebank import TreebankWordDetokenizer
5 from .models import Bloom, Verb, Conjugation
6 from collections import defaultdict
7 import re
8 import json
9
10 def analyze_nltk(my_text):
11     detokenize = TreebankWordDetokenizer().detokenize
12     verbs = list(Verb.objects.values_list('name', flat=True))
13     conj = list(Conjugation.objects.values_list('name', flat=True))
14     tokenized = word_tokenize(my_text)
15     words_found = []
16     result = defaultdict(list)
17     for index, word in enumerate(tokenized):
18         if word in verbs or word in conj:
19             words_found.append(word)
20             tokenized[index] = "<mark><strong>" + word + "</strong></mark>"
21     text_highlighted = detokenize(tokenized)
22     if words_found:
23         conj_found = Conjugation.objects.filter(name__in=words_found)
24         for conj in conj_found:
25             bloom_sections = conj.base_verb.bloom_section.all()
26             for bloom in bloom_sections:
27                 name = bloom.name
28                 result[name].append(conj.name)
29     result = dict(result)
30     context = {
31         'text': text_highlighted,
32         'words_found': result
33     }
34     return context
35
```

Figura 3.1: Solución a través de código para la localización de verbos.

visualizar de una forma sencilla el funcionamiento del sistema de software y abarca desde que se ingresa un archivo de tipo texto, hasta la salida que debe mostrar. Tal como se visualiza en la Figura 3.2:

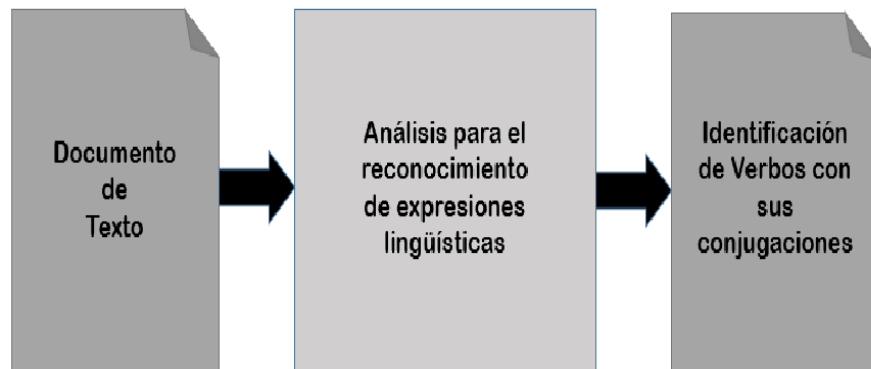
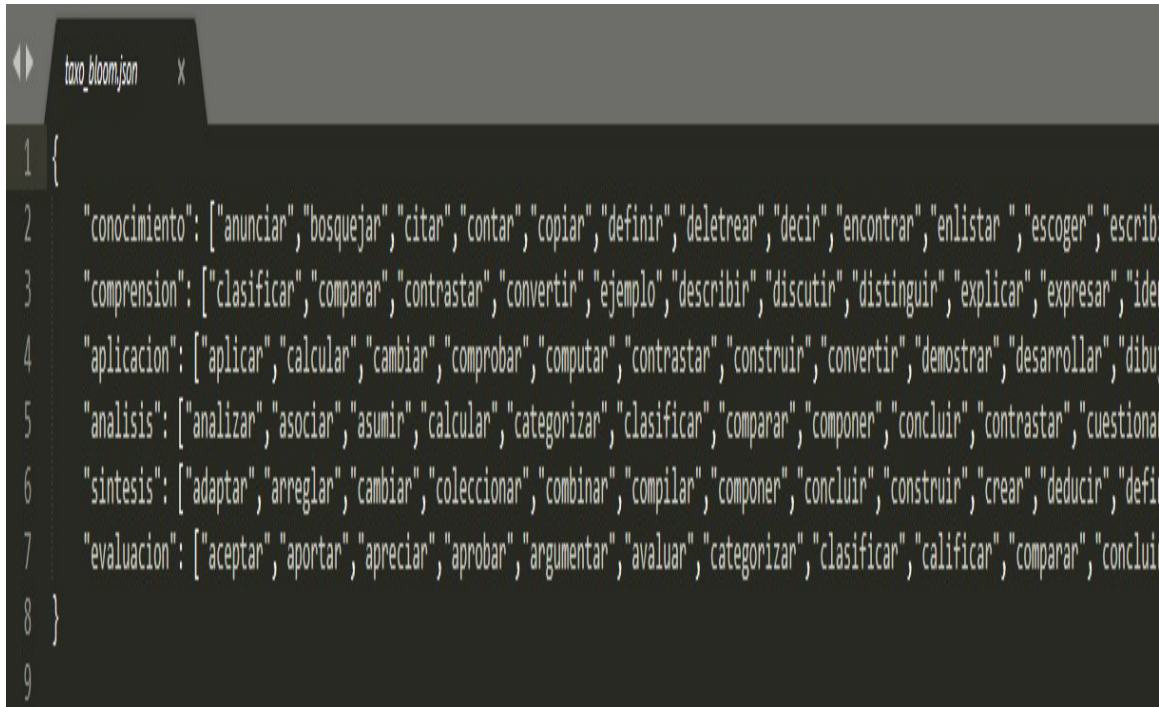


Figura 3.2: Diagrama de casos de uso general.

3.3. Construcción del prototipo

En esta sección se explicarán partes fundamentales del sistema y a su vez se mostrará el código que sustente dicha explicación.

Primero que nada, se construye la tabla o referencia, en la que estarán definidos los verbos por categorías. De esta manera al ingresar un texto y correr el script, se tiene un contenido como base para saber que palabras (en este caso verbos) debe identificar y arrojar como resultado al momento de ejecutarlo. Dicho argumento se presenta en la Figura 3.3:



```

1 {
2   "conocimiento": ["anunciar", "bosquejar", "citar", "contar", "copiar", "definir", "deletrear", "decir", "encontrar", "enlistar", "escoger", "escribi
3   "comprension": ["clasificar", "comparar", "contrastar", "convertir", "ejemplo", "describir", "discutir", "distinguir", "explicar", "expresar", "ide
4   "aplicacion": ["aplicar", "calcular", "cambiar", "comprobar", "computar", "contrastar", "construir", "convertir", "demostrar", "desarrollar", "dibu
5   "analisis": ["analizar", "asociar", "asumir", "calcular", "categorizar", "clasificar", "comparar", "componer", "concluir", "contrastar", "cuestionar
6   "sintesis": ["adaptar", "arreglar", "cambiar", "coleccionar", "combinar", "compilar", "componer", "concluir", "construir", "crear", "deducir", "defi
7   "evaluacion": ["aceptar", "aportar", "apreciar", "aprobar", "argumentar", "avaluar", "categorizar", "clasificar", "calificar", "comparar", "conclui
8 }
9

```

Figura 3.3: Imagen de la definición de Taxonomía de Bloom en código.

Posteriormente para la identificación de los verbos que comprende la Taxonomía de Bloom en su singular categorización, y que a su vez es lo esencial del proyecto desarrollado, se generó un script nombrado `Helpers` y en él se realiza lo siguiente: se importa el modelo o base de verbos que se tiene y que corresponde a: `from verbs.models import Bloom, Verb, Conjugation`. También se importan los métodos de NLTK que son `word_tokenize` y que ayudará a Tokenizar el texto, es decir desarmará el string en palabras creando un arreglo. Se importa la clase que ayudara a destokenizarlo o convertir nuevamente en string. Posteriormente se importa la librería de `pattern` para poder hacer posible la conjugación de los verbos: `from pattern.es import conjugate as conj`.

Enseguida se construye una tabla, con las opciones de conjugación que se debe aplicar para el modelo de verbos que ya se tiene.

Posteriormente se procede a analizar el texto, en el cual se indica resaltar los verbos localizados en sus conjugaciones, analizando a la par a cuál de las secciones de la Taxonomía de Bloom corresponde. Para visualizarlo rápidamente se muestra la Figura 3.4:

```

3  from nltk import word_tokenize  ##Importamos solo la funcion word_tokenize de la libreria
4  from nltk.tokenize.treebank import TreebankWordDetokenizer
5  from .models import Bloom, Verb, Conjugation
6  from collections import defaultdict
7  import re
8  import json
9
10 def analyze_nltk(my_text):
11     detokenize = TreebankWordDetokenizer().detokenize
12     verbs = list(Verb.objects.values_list('name', flat=True))
13     conj = list(Conjugation.objects.values_list('name', flat=True))
14     tokenized = word_tokenize(my_text)
15     words_found = []
16     result = defaultdict(list)
17     for index, word in enumerate(tokenized):
18         if word in verbs or word in conj:
19             words_found.append(word)
20             tokenized[index] = "<mark><strong>" + word + "</strong></mark>"
21     text_highlighted = detokenize(tokenized)
22     if words_found:
23         conj_found = Conjugation.objects.filter(name__in=words_found)
24         for conj in conj_found:
25             bloom_sections = conj.base_verb.bloom_section.all()
26             for bloom in bloom_sections:
27                 name = bloom.name
28                 result[name].append(conj.name)
29     result = dict(result)
30     context = {
31         'text': text_highlighted,
32         'words_found': result
33     }
34     return context

```

Figura 3.4: Código para la identificación de verbos en sus conjugaciones.

3.4. Evaluación del prototipo por el cliente

Se realizaron pruebas con documentos de tipo texto de entrevistas reales proporcionadas por el cliente. Al analizar dichos documentos el sistema de software si logra identificar aquellos verbos que se encuentran dentro del mismo y en todas sus conjugaciones. Para corroborar lo antes mencionado se muestra a continuación la Figura 3.5 como evidencia de una de las pruebas elaboradas:

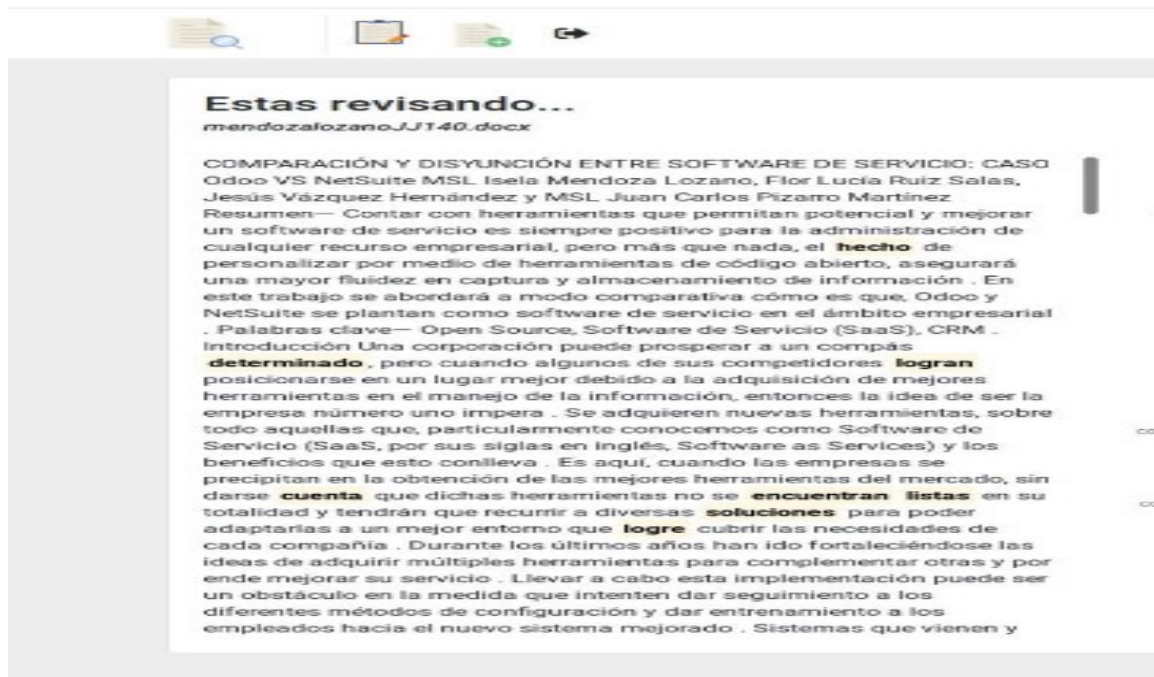


Figura 3.5: Pantalla de identificación de verbos.

3.5. Refinamiento del prototipo

Se han mostrado los resultados a quien funge como cliente y proporcionó más entrevistas reales para el análisis y comprobación de resultados. Posteriormente calificó como satisfactoria la ejecución del sistema, solicitando se hicieran ligeros cambios de lo que se tenía estipulado en un principio y que se detalla a continuación:

Lo estipulado era que el sistema constara de una única pantalla, la cual se utilizaría siempre que se deseara hacer la revisión sintáctica de una entrevista escrita. Dicha pantalla se muestra a continuación en la Figura 3.6:

En dicha pantalla se muestra un espacio en el cual se debe arrastrar el documento escrito a analizar o bien, se puede seleccionar desde el botón etiquetado como “subir documento”. Posteriormente se proporciona un menú de 5 opciones con check box para que se seleccionen



Figura 3.6: Pantalla de revisión sintáctica.

los puntos a revisar del documento, estas opciones son:

- Seleccionar Todo
- Identificación de Creencias
- Requisitos Indispensables
- Requisitos Opcionales
- Verbos Taxonomía de Bloom

Después de seleccionar una o varias de las opciones mencionadas, se presionará el botón de “analizar documento”. Posterior a esto, en el espacio en blanco que aparece a la derecha, se mostrará el texto a analizar con las características seleccionadas resaltadas, es decir, si se seleccionó la opción de identificación de creencias, las palabras o frases que se identifiquen

como tal se subrayaran con el color verde (por dar un ejemplo), los requisitos opcionales de otro color y así sucesivamente.

Los cambios solicitados son los siguientes:

1. Eliminar todos los botones, excepto el de verbos de Taxonomía de Bloom, pero continuando con el mismo objetivo: identificar los verbos de la Taxonomía de Bloom en todas sus conjugaciones.

2. Aunado a esto, por medio de un análisis de las entrevistas proporcionadas por el cliente, se solicitó elaborar una Taxonomía orientada a la ingeniería de requisitos, dando con ello un plus y una herramienta de basta ayuda para dicha disciplina.

3. Por último, el sistema dejó de ser para interfaz de escritorio y se implementó para web, por las siguientes razones:

- El programa no se instala directamente en el equipo.
- Los documentos estarán almacenados en un solo sitio.
- Se podrá acceder al sistema desde cualquier punto para poder realizar un análisis.

3.6. Producto de ingeniería

Actualmente el sistema se encuentra funcional y terminado, respetando las modificaciones anteriormente mencionadas y teniendo la siguiente funcionalidad:

Primero que nada, para poder ingresar al sistema se solicita un usuario y contraseña, posteriormente se encuentran iconos, uno de ellos muestra el listado de verbos que comprende la Taxonomía de Bloom, los muestra de 12 en 12 y si se presiona uno de los verbos, nuevamente muestra de 12 en 12 sus conjugaciones. A continuación se muestra en la Figura 3.7 la pantalla de autenticación de usuario para el sistema de software:

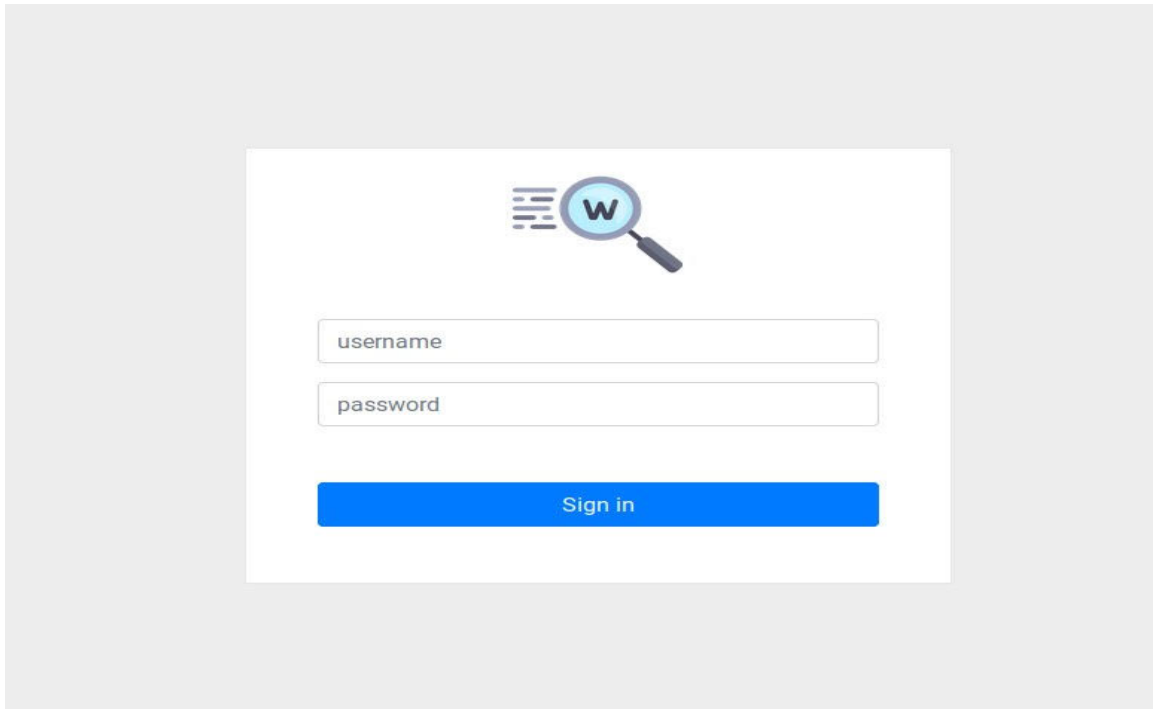


Figura 3.7: Pantalla de autenticación de usuario.

Otro de los botones es el que permite seleccionar el archivo a analizar. Una vez que se sube el archivo los verbos localizados se resaltan para visualizarlos fácilmente. En este mismo escenario, a la derecha, se muestra una gráfica en la cual se aprecia la cantidad de verbos localizados en el documento, ya clasificados según la categorización de la Taxonomía de Bloom, tal como se muestra en la Figura 3.8:

En este mismo botón, antes de seleccionar un archivo se muestran los documentos analizados con anterioridad por medio de una tabla que consta de tres columnas; en la primera muestra el nombre del archivo, en la segunda el nombre del usuario que analizo dicho documento y por último la fecha y la hora en que se dio el evento y se puede visualizar a continuación en la Figura 3.9:

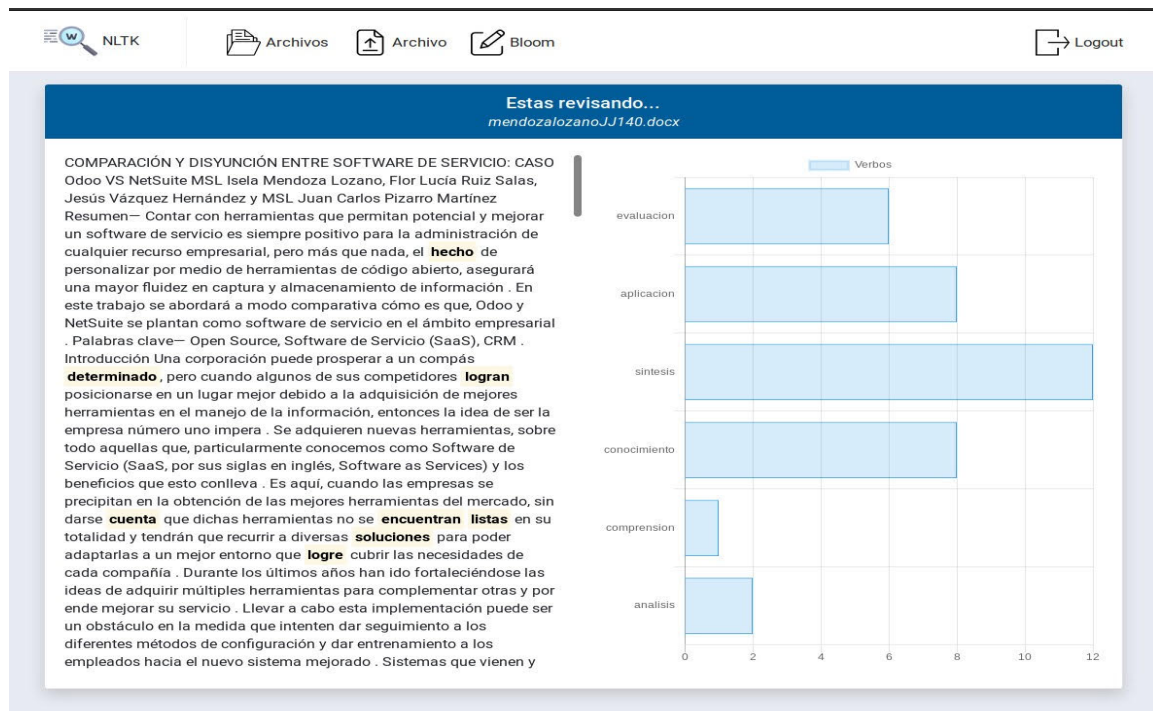


Figura 3.8: Pantalla de identificación de verbos y graficación.

3.7. Instalación de complementos

Se instaló un entorno de desarrollo para visualizar de una manera más ordena las líneas de código generadas, dicho entorno se denomina Sublime Text 3. Este, a su vez, se predeterminó para que la separación sintáctica fuera acorde a Python tal como se muestra en la Figura 3.10:

Otro complemento que se instaló fue Django, que funge como la parte de servidor para poder visualizar en la web el proyecto planteado. Dentro del mismo se instalan cada uno de los complementos a utilizar, tales como: docx2txt, Pattern, NLTK, así como también se corre un comando `python manage.py runserver` para poder tener un local host. Las indicaciones mencionadas se realizan dentro de la carpeta de entorno virtual en la que se trabajó, la cual, a su vez se encuentra dentro de la carpeta en donde se guardó el proyecto. Todo esto, se hace

Archivos			
File name	User name	Created at	Actions
update_service_ttco_db.txt	vane	Aug. 10, 2019, 2:49 a.m.	
prueba_nltk.docx	jvh	Aug. 10, 2019, 1:50 a.m.	
leia_fix.txt	jvh	July 9, 2019, 1:04 a.m.	
mendozalozanoJJ140.docx	jvh	July 8, 2019, 8:20 p.m.	

1 of 1

Figura 3.9: Pantalla de la lista de archivos analizados.

en consola, tal como se muestra en la siguiente Figura 3.11:

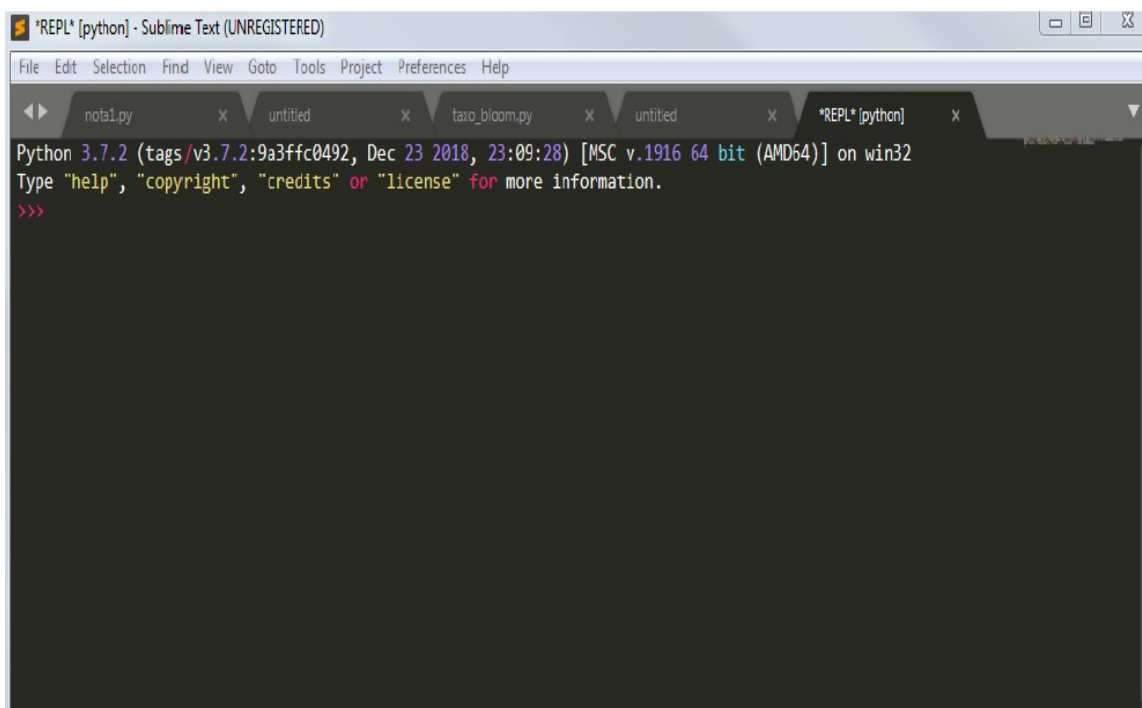
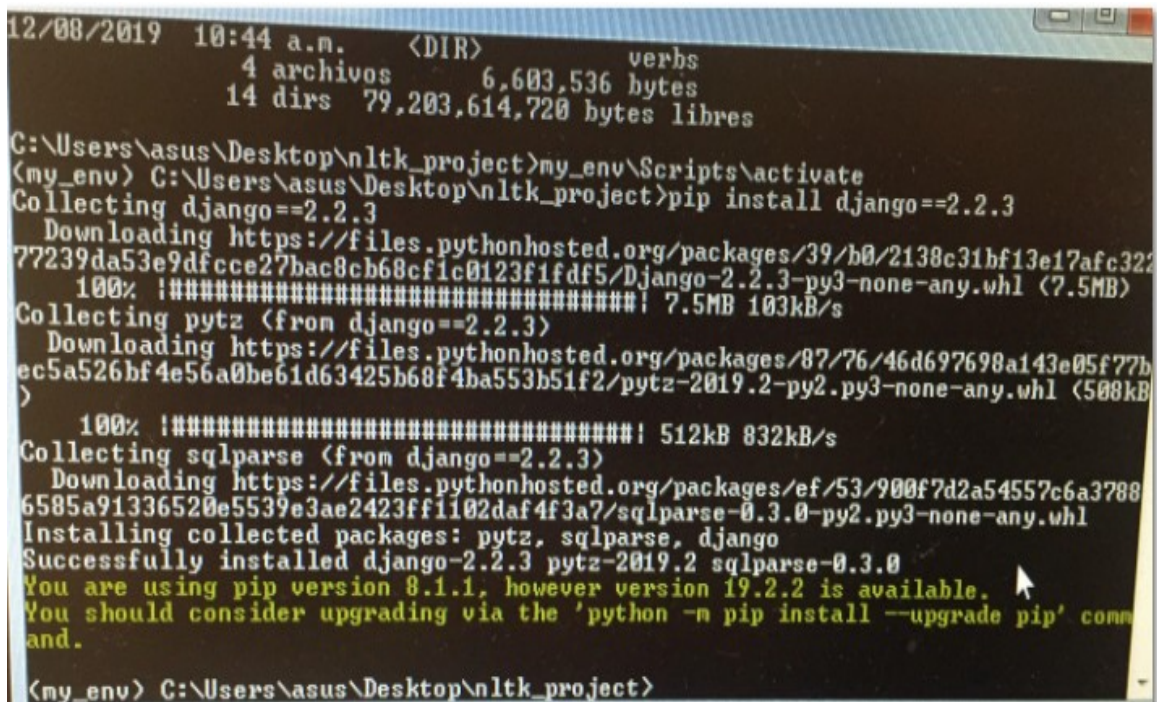


Figura 3.10: Pantalla que muestra el entorno de desarrollo Sublime Text 3.



```
12/08/2019 10:44 a.m. <DIR> verbs
4 archivos 6,603,536 bytes
14 dirs 79,203,614,720 bytes libres

C:\Users\asus\Desktop\nltk_project>my_env\Scripts\activate
(my_env) C:\Users\asus\Desktop\nltk_project>pip install django==2.2.3
Collecting django==2.2.3
  Downloading https://files.pythonhosted.org/packages/39/b0/2138c31bf13e17afc322
77239da53e9dfcce27bac8cb68cf1c0123f1fdf5/Django-2.2.3-py3-none-any.whl (7.5MB)
    100% |#####| 7.5MB 103kB/s
Collecting pytz (from django==2.2.3)
  Downloading https://files.pythonhosted.org/packages/87/76/46d697690a143e05f77b
ec5a526bf4e56a0be61d63425b68f4ba553b51f2/pytz-2019.2-py2.py3-none-any.whl (508kB)
    100% |#####| 512kB 832kB/s
Collecting sqlparse (from django==2.2.3)
  Downloading https://files.pythonhosted.org/packages/ef/53/900f7d2a54557c6a3788
6585a91336520e5539e3ae2423ff1102daf4f3a7/sqlparse-0.3.0-py2.py3-none-any.whl
Installing collected packages: pytz, sqlparse, django
Successfully installed django-2.2.3 pytz-2019.2 sqlparse-0.3.0
You are using pip version 8.1.1, however version 19.2.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.

(my_env) C:\Users\asus\Desktop\nltk_project>
```

Figura 3.11: Pantalla que muestra la instalación de Django.

Capítulo 4

Resultados y discusiones

En este capítulo se mencionarán los resultados obtenidos después del desarrollo del sistema de software que se ha venido mencionando a lo largo de este documento. También se discutirá la evaluación por la que atravesó el sistema para determinar si era lo esperado y si el manejo del mismo es sencillo o demasiado complicado para el cliente.

En este capítulo se mostrará un ejemplo de las entrevistas analizadas, por cuestiones de confidencialidad solo va a reflejarse una parte de la misma. Tal como se muestra en la siguiente Figura 4.1:

En la figura anterior se puede apreciar como el sistema de software es capaz de identificar los verbos conjugados y lo muestra resaltándolos por colores, según el nivel cognitivo de la clasificación de la Taxonomía de Bloom. Enseguida, a la derecha se puede observar cómo es que se genera una gráfica de barras en la que indica cuantos y cuales verbos se identificaron en cada una de las categorizaciones de la Taxonomía de Bloom.

Esta vista demuestra que el objetivo principal del desarrollo de este proyecto se cumple, ya que si realiza la función de identificar los verbos conjugados y analizar textos (para este caso entrevistas) en un corto tiempo.

A su vez se tiene otro resultado visible del proyecto y es el reporte que se genera y se puede descargar después del análisis de una entrevista. Dicho reporte se descarga en Excel y



Figura 4.1: Pantalla que muestra el análisis de una entrevista

en el muestra el nombre del archivo que se analizó, las categorías de la Taxonomía de Bloom y dentro de cada una los verbos que a lo largo del texto aparecieron y que corresponde a las mismas. El reporte se ve como se muestra en la Figura 4.2:

De manera posterior se generó una tabla de Taxonomía de Bloom orientada a ingeniería de requisitos, esta se realizó después de analizar varias entrevistas de casos reales que el cliente proporciono, según la localización de verbos fue como se organizó la tabla, respetando la clasificación original, ya que lo que se busca mostrar es la incidencia de verbos por categoría dentro de la disciplina de ingeniería de requisitos. Para visualizarlo de una mejor manera enseguida se presentan las Figuras 4.3 y 4.4:

Para concluir se puso a prueba el sistema de software con ingenieros de requisitos en entrevistas de casos reales, se aplicó un examen de usabilidad y en general se cataloga como

	A	B	C	D	E	F	G	H	I	J	K	L
1	Nombre del documento:											
2	Entrevistas de audio a texto.docx											
3												
4												
5	Sección	Palabras encontradas										
6	aplicacion	cambio	muestra	muestro	operamos	usa	usado	usamos	usas	utilizamos		
7	synthesis	cambio	creando	creas	creo	crea ³	definimos	escriben	escribo	establece	hace	hacemos
8	conocimiento	contando	decimos	dec ³ -a	definimos	dice	dicen	diciendo	digas	digo	dijo	escriben
9	analysis	diferencia	funciona									
10	comprensión	identifica	revisan	revisando	seleccionas	selecciono						
11	evaluación	marcan	seleccionas	selecciono								
12												
13												

Figura 4.2: Pantalla que muestra el reporte de verbos en una entrevista

un sistema con una interfaz amigable para el usuario, fácil de manejar y de gran utilidad para el área de ingeniería de requisitos. Para demostrar lo mencionado se muestran la Figura 4.5 y 4.6:

← Procesos cognitivos de orden inferior →			← Procesos cognitivos de orden superior →		
Recordar / Conocimiento	Comprender	Aplicar	Analizar	Evaluar	Crear / Síntesis
Definiendo Dice Identifica Identifican Nombre Cuenta Decimos Dicen Diciendo Diga Digamos Digo Dijo Identificamos Muestra Nombre Registra Registrado Registro Cita Dicho Etiquetas Reconoce Recordando Lista Contando Definimos Digas Escriben Escribo Muestro Seleccionas Selecciono	Identifica Identifican Interpretando Convierte Identificamos Revisado Revisamos Revisando Resumiendo Interpreta Revisan Seleccionas Selecciono	Interpretación Programas Utilizando Aplica Cambiamos Cambiaste Cambio Convierte Modelo Modificamos Muestra Programa Resuelve Usamos Usan Utiliza Utilizamos Aplicara Interpreta Usa Muestro Operamos Usado Usas	Analicemos Clasifican Cuestiones Diferencia Funcionando Descubriendo Dividirá Funciona	Considera Decidiendo Determinado Clasifican Interpretando Considera Decide Decidimos Decido Determina Determinan Significa Descubriendo Interpreta Logre Marcan	Definiendo Haciendo Hecho Interpretando Hacemos Arreglamos Arreglen Cambiamos Cambiaste Cambio Creo Especificado Especifico Hace Hacemos Hacen Haces Haciendo Hago Hicieron Hizo Modificamos Resuelve Hiciera Hicieran Hicieron Interpreta Propone Sucede Sucedido Creando Creas Crea Definimos Escriben Escribo Establece Haces Hice Hicimos Sucedió Supne

Figura 4.3: Taxonomía de Bloom para ingeniería de requisitos (verbos frecuentes)

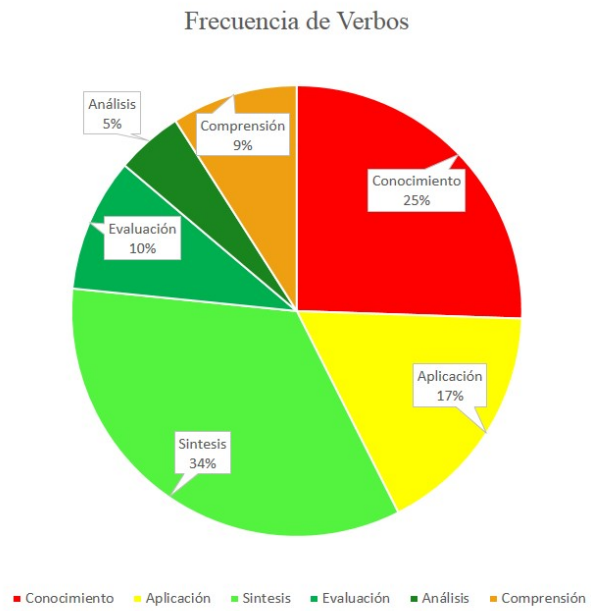


Figura 4.4: Gráfica con incidencia de verbos por categoría

Examen de medición de usabilidad de software

¿Para qué, en general, utiliza este software?

Para identificar necesidades de mis clientes al momento de entrevistarlos.

Preguntas	De acuerdo	Indeciso	Desacuerdo
Este software responde muy lentamente a las entradas.			X
Recomendaría este software a mis colegas	X		
Las Instrucciones e indicaciones son útiles		X	
Este software se ha detenido inesperadamente en algún momento			X
Aprender a operar este software inicialmente está lleno de problemas.			X

Figura 4.5: Examen de usabilidad

¿Qué importancia tiene para usted el tipo de software que acaba de calificar?

- a) Extremadamente importante
- b) Importante
- c) No es muy importante
- d) Nada importante

¿Cómo calificaría sus habilidades y conocimientos de software?

- a) Muy experimentado y técnico
- b) Soy experimentado pero no técnico
- c) Puedo hacer frente a la mayoría del software
- d) Encuentro que la mayoría del software es difícil de usar

¿Cree usted que este software le ayuda a reducir el tiempo de análisis de una entrevista? ¿Por qué?

Si, ya que me permite indagar necesidades al tiempo de analizar

¿Cuál crees que es el mejor aspecto de este software y por qué?

El mostrar en el editor de texto los verbos en forma de resaltado y categorizado por tipos de verbos en una barra.

Figura 4.6: Examen de usabilidad

Capítulo 5

Conclusiones

A lo largo del desarrollo del proyecto propuesto en este documento, se ha venido cuidando un objetivo principal y lineamientos que encaminaron a lograrlo. En este capítulo se darán a conocer las conclusiones que se tienen acerca del proyecto, de los resultados que se obtuvieron una vez que se culminó el sistema de software y de los resultados que determinaron los ingenieros de requisitos que evaluaron el funcionamiento del mismo. También se han de mencionar mejoras futuras que se encontraron como factibles para este sistema de software.

5.1. Con respecto al objetivo de la investigación

Haciendo una breve recapitulación, el objetivo principal es identificar el conocimiento tácito a través de la Taxonomía de Bloom, con el fin de reducir el tiempo de análisis en las entrevistas de elicitación de requisitos, empleando el procesamiento de lenguaje natural mediante un sistema de software para reconocer expresiones lingüísticas.

Dicho objetivo se encuentra como cumplido, ya que el sistema de software desarrollado cubre los aspectos de identificar conocimiento tácito a través de la identificación de los verbos que se encuentran dentro de las entrevistas analizadas. Lo anteriormente mencionado lo hace teniendo como base la Taxonomía de Bloom para la categorización de los verbos. Se hizo uso de lenguaje natural y Pattern fue la librería principal, empleada precisamente para la

identificación de los verbos en sus diferentes conjugaciones.

La principal aportación para el área de ingeniería de requisitos con el desarrollo de este sistema de software, era el acortar el tiempo en el que ellos hacen el análisis de una entrevista para determinar qué es lo que realmente su cliente necesita. Hoy, se puede decir que el propósito se cumple y si es de gran aportación el sistema desarrollado, ya que en cuestión de 2 minutos, desde que se inicia sesión hasta que se sube el archivo de texto a analizar, se pueden tener identificados los verbos por colores según su nivel cognitivo y así mismo se puede obtener un reporte en el que se muestren los verbos localizados, agrupados por categoría.

5.2. Recomendaciones para futuras investigaciones

Como recomendaciones para futuras investigaciones sobre el tema que se expone en este documento se lograron identificar las siguientes:

- Se le pueden agregar más identificadores lingüísticos al sistema de software, tales como presuposiciones, de los cuales también pudiera generarse un reporte.
- La clasificación de requisitos es otra importante implementación que puede hacerse a futuro.
- En el desarrollo de este sistema de software se intentó utilizar la librería NLTK, la cual no funcionó para hacer la conjugación de verbos, en un futuro puede ser que ya tenga más funcionalidades.
- Otra importante recomendación es que después de analizar en el sistema de software desarrollado, un considerable número de entrevistas de casos reales, se puede trabajar en como diseñar entrevistas. Esto a raíz de que el análisis puede mostrar que tipos de verbos son más frecuentes en las respuestas de preguntas comunes.

Es así como concluye el desarrollo de la investigación y construcción del sistema para el

reconocimiento de expresiones lingüísticas en entrevistas para ingeniería de requisitos.

Bibliografía

- [1] K. Olmos and J. Rodas, “KMoS-RE: knowledge management on a strategy to requirements engineering”, *Requirements Engineering*, vol. 19, no. 4, pp. 421–440, oct 2014.
- [2] Domingo Valhondo, “Gestión del conocimiento: del mito a la realidad”, 2002.
- [3] I. Nonaka, “La empresa creadora de conocimiento”, Tech. Rep., 2007.
- [4] Berenice Ing Torres, Karla Dra Olmos, Jorge Dr Rodas, and Rogelio Dr Florencia, “KRIM: Modelo para la Identificación de Conocimiento Tácito y Clasificación de requisitos”, Tech. Rep., 2015.
- [5] F. Edgar Castillo-Barrera, Monica Amador-Garcia, Hector G. Perez-Gonzalez, Francisco E. Martinez-Perez, and Francisco J. Torres-Reyes, “Adapting bloom’s taxonomy for an agile classification of the complexity of the user stories in SCRUM”, in *Proceedings - 2018 6th International Conference in Software Engineering Research and Innovation, CONISOFT 2018*. feb 2019, pp. 139–145, Institute of Electrical and Electronics Engineers Inc.
- [6] Leonardo Campillos Llanos, “Tecnologías del habla y análisis de la voz. Aplicaciones en la enseñanza de la lengua *”, 2010.
- [7] Yongchuan Tang, “A collective decision model involving vague concepts and linguistic expressions”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 2, pp. 421–428, apr 2008.

- [8] J ; Ochoa, “Clasificación Automática de Foros de Discusión de Acuerdo al Dominio Cognitivo de la Taxonomía de Bloom Empleando Minería de Texto y un Clasificador Bayesiano”, Tech. Rep., 2013.
- [9] R. M. Rodríguez, Luis Martínez, and Francisco Herrera, “Eliciting comparative linguistic expressions in group decision making”, in *Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting, IFSA/NAFIPS 2013*, 2013, pp. 771–776.
- [10] Horsella y Nelson Allendes, M, “EXPRESIONES LINGÜÍSTICAS EN EL DISCURSO ARGUMENTATIVO: UNA APROXIMACIÓN SEMÁNTICA”, Tech. Rep., 1995.
- [11] A Warde, “Tacit and Explicit Knowledge By Harry Collins”, *Sociological Review -Keele-*, vol. 58, no. 4, pp. 714–716, 2010.
- [12] Charles Lowney, “Ineffable, Tacit, Explicable and Explicit”, *Tradition and Discovery: The Polanyi Society Periodical*, vol. 38, no. 1, pp. 18–37, 2011.
- [13] M. A. Méndez Oramas, “La taxonomía de Bloom, una herramienta imprescindible para enseñar y aprender » CENTRO DEL PROFESORADO Tenerife Sur”, 2015.
- [14] Isidoro Hernán-Losada, Carlos A Lázaro-Carrascosa, J Ángel Velázquez-Iturbide, Universidad Rey, and Juan Carlos, “Una Aplicación Educativa Basada en la Jerarquía de Bloom para el Aprendizaje de la Herencia de POO”, Tech. Rep., 2005.
- [15] Robert Stalnaker, “Presuppositions”, in *Contemporary Research in Philosophical Logic and Linguistic Semantics*, pp. 31–41. Springer Netherlands, Dordrecht, 1975.
- [16] Franklin Sentis, “LA PRESUPOSICIÓN COMO CATEGORÍA PRAGMÁTICA: ... 105 LA PRESUPOSICIÓN COMO CATEGORÍA PRAGMÁTICA: UN CASO DE CONFRONTACIÓN EPISTEMOLÓGICA”, Tech. Rep., Onomázein, 2001.
- [17] L. Karttunen, “Presuposición y contexto lingüístico”, 1998.

- [18] P Pytel, C Uhalde, H Ramón, H Castello, and M Tomasello, “INGENIERÍA DE REQUISITOS BASADA EN TÉCNICAS DE INGENIERÍA DEL CONOCIMIENTO”, Tech. Rep., Argentina, 2011.
- [19] Mg Augusto Cortez Vásquez, Mg Hugo Vega Huerta, Lic Jaime, and Pariona Quispe, “Procesamiento de lenguaje natural”, *Revista de Investigación de Sistemas e Informática (RISI)*, vol. 6, no. 2, 2009.
- [20] Charley. Lecht, *Informacion Tsunami: Un futurista mira en retrospectiva*, Limusa, México D.F, 1992.
- [21] A. Moreno, “Procesamiento del lenguaje natural ¿qué es? - IIC”, 2017.
- [22] Antonio Moreno, “Procesamiento del lenguaje natural ¿qué es? - IIC”.
- [23] Richard M Reese, *Natural Language Processing with Java*, Packt Publishing Ltd, 2015.

Apéndice A

Tabla de la Taxonomía de Bloom

En esta sección se incluye la tabla de la Taxonomía de Bloom para poder visualizarla a detalle.

Procesos cognitivos de orden inferior			Procesos cognitivos de orden superior		
RECORDAR	COMPRENDER	APLICAR	ANALIZAR	EVALUAR	CREAR
Recordar hechos/datos sin necesidad de entender. Se muestra material aprendido previamente mediante el recuerdo de términos, conceptos básicos y respuestas.	Mostrar entendimiento a la hora de encontrar información del texto. Se demuestra comprensión básica de hechos e ideas.	Usar en una nueva situación. Resolver problemas mediante la aplicación de conocimiento, hechos o técnicas previamente adquiridas en una manera diferente.	Examinar en detalle. Examinar y descomponer la información en partes identificando los motivos o causas; realizar inferencias y encontrar evidencias que apoyen las generalizaciones.	Justificar. Presentar y defender opiniones realizando juicios sobre la información, la validez de ideas o la calidad de un trabajo basándose en una serie de criterios.	Cambiar o crear algo nuevo. Recopilar información de una manera diferente combinando sus elementos en un nuevo modelo o proponer soluciones alternativas.
PALABRAS CLAVE:	PALABRAS CLAVE:	PALABRAS CLAVE:	PALABRAS CLAVE:	PALABRAS CLAVE:	PALABRAS CLAVE:
Elegir Copiar Definir Decir Citar Leer Quién Recitar Cómo Por qué	observar omitir rastrear cuándo repetir relacionar listar escribir localizar dónde Memorizar	mostrar deletrear afirmar duplicar ilustrar demostrar discutir revisar mostrar resumir observar	Preguntar Generalizar Clasificar Comparar Contrastar Parafrasear Informar Inferir Interpretar Explicar Expresar Traducir	esquematizar predecir dar ejemplos relacionar ilustrar demostrar discutir revisar mostrar resumir observar	Actuar emplear practicar Identificar seleccionar agrupar Calcular elegir resumir Entrevistar planear desarrollar Enseñar transferir interpretar Usar demostrar categorizar Conectar dramatizar construir Planear manipular resolver Simular seleccionar Hacer uso organizar
Examinar Centrarse Razonar Inferencia Comparar Dividir Buscar Inspeccionar Simplificar Preguntar Elegir Establecer Encuesta	priorizar encontrar asumir destacar causa-efecto aislar separar reorganizar motivar diferenciar descomponer Investigar categorizar ordenar poner a prueba observar	Medir opinar argumentar Evaluar premiar testar Decidir debatir convencer Apoyar explicar seleccionar Defender comparar deducir Justificar percibir recomendar Crítico probar estimar Juzgar influir persuadir Valorar demostrar	Adaptar añadir estimar planear Añadir experimentar testar Construir extender sustituir Cambiar formular reescribir Combinar hipotetizar suponer Componer innovar teorizar Compilar mejorar pensar Componer maximizar simplificar Crear minimizar proponer Descubrir modelar visualizar Diseñar modificar Desarrollar originar Elaborar transformar		
ACCIONES	RESULTADO	ACCIONES	RESULTADO	ACCIONES	RESULTADO
Describir Encontrar Identificar Listar Localizar Nombrar Reconocer Recuperar	Definición Hechos Etiquetado Listado Cuestionario Interpretación Test Cuaderno Fotocopia	Clasificar Comparar Ejemplificar Explicar Inferir Interpretar Parafrasear Resumir	Colección Ejemplos Explicación Etiquetado Listado Esquema Cuestionario Resumen Muestra y cuenta	Desempeñar Ejecutar Implementar Usar Emplear Realizar	Demostración Diario Ilustraciones Entrevista Interpretación Simulación Presentación Dibujo
Atribuir Deconstruir Integrar Organizar Esquematizar Estructurar Encuesta Hoja de cálculo	Reseña Gráfica Lista de control Base de datos Gráfico Informe Encuesta Hoja de cálculo	Atribuir reseña Comprobar gráfica base de datos Informe hoja de cálculo encuesta Estructurar	Construir anuncio película Trazar juego dibujar plan proyecto canción Historia Producto audiovisual		
PREGUNTAS	PREGUNTAS	PREGUNTAS	PREGUNTAS	PREGUNTAS	PREGUNTAS
¿Puedes enumerar...? ¿Puedes recordar...? ¿Puedes seleccionar...? ¿Cómo ocurrió...? ¿Cómo es...? ¿Cómo describirías...? ¿Podrías explicar...? ¿Cómo mostrarías...? ¿Qué es...? ¿Cuál...? ¿Quién fue...? ¿Quiénes fueron los principales...? ¿Por qué...?	¿Puedes explicar que está ocurriendo...? ¿Cómo clasificarías...? ¿Cómo compararías/contrastarías...? ¿Cómo podrías parafrasear el significado de...? ¿Cómo resumirías...? ¿Qué puedes decir sobre...? ¿Cuál es la mejor respuesta...? ¿Qué afirmaciones apoyan...? ¿Podrías afirmar o interpretar en tus propias palabras...?	¿Cómo usarías...? ¿Qué ejemplos sobre...puedes encontrar? ¿Cómo organizarías... para presentar...? ¿Cómo aplicarías lo que has aprendido para desarrollar...? ¿Qué enfoque usarías para...? ¿Qué aspectos seleccionarías para mostrar...? ¿Qué preguntas harías en una entrevista a...?	¿Cuáles son las partes o rasgos de...? ¿En qué aspectos está...? ¿Relacionado/a con...? ¿Por qué opinas que...? ¿Qué motivo hay para...? ¿Puedes hacer un listado de las partes...? ¿Qué ideas justifican...? ¿Qué conclusiones extraes de...? ¿Qué evidencias de... encuentras? ¿Puedes distinguir entre...? ¿Cuál es la relación entre...? ¿Cuál es la función de...?	¿Estás de acuerdo con...? ¿Cuál es tu opinión sobre...? ¿Cómo comprobarías...? ¿Sería mejor si...? ¿Por qué ese personaje...? ¿Cómo valorarías...? ¿Cómo determinarías...? ¿Cómo priorizarías...? ¿Qué información podrías para apoyar tu punto de vista? ¿Cómo justificarías...? ¿Qué datos te llevaron a esa conclusión? ¿Qué seleccionarías para...? ¿Qué elección hubieras tomado si...?	¿Qué cambios harías para...? ¿Cómo mejorarías...? ¿Qué pasaría si...? ¿Podrías proponer una alternativa? ¿Puedes elaborar...basándote en...? ¿De qué forma evaluarías...? ¿Podrías formular una teoría alternativa? ¿Qué harías para maximizar/minimizar...? ¿Cómo pondrías a prueba...? ¿Podrías construir un modelo que cambie...? ¿Se te ocurre un modo original para...? ¿Cómo cambiarías el guión/plan? ¿Cómo adaptarías... para...?

Figura A.1: Tabla de la Taxonomía de Bloom.