UNIVERSIDAD DEL SINÚ
Elías Bechara Zainúm
Seccional Cartagena

ANÁLISIS COMPARATIVO DE DOCUMENTOS FRENTE A OBJETIVOS ESTABLECIDOS USANDO ANALÍTICA DE TEXTO:

Caso de estudio Syllabus de los cursos de Unisinú

EDER ORTEGA CARDONA
ALDAIR CANO YANEZ

UNIVERSIDAD DEL SINÚ
ESCUELA DE INGENIERÍA DE SISTEMAS
CARTAGENA DE INDIAS D.T. Y C.
2018

UNIVERSIDAD DEL SINÚ
Elías Bechara Zainúm
Seccional Cartagena

ANÁLISIS COMPARATIVO DE DOCUMENTOS FRENTE A OBJETIVOS ESTABLECIDOS USANDO ANALÍTICA DE TEXTO:
Caso de estudio Syllabus de los cursos de Unisinú

EDER ORTEGA CARDONA
ALDAIR CANO YANEZ

Trabajo de grado para optar por el título de Ingeniero de Sistemas

Asesor disciplinar
RAFAEL MONTERROZA
Asesor metodológico
EUGENIA ARRIETA RODRIGUEZ

UNIVERSIDAD DEL SINÚ
ESCUELA DE INGENIERÍA DE SISTEMAS
CARTAGENA DE INDIAS D.T. Y C.

ACTA DE CALIFICACION Y APROBACION

Nota de aceptación:

_____

_____

_____

_____

_____

Director de Escuela

_____

Director de Investigaciones

_____

Firma del jurado

_____

Firma del jurado

Cartagena de Indias, 14 de Noviembre de 2018

Cartagena de Indias, 14 de Noviembre de 2018

Director
Maria Claudia Bonfante
Director de la Escuela de Ingeniería de Sistemas
Universidad del Sinú

Cordial saludo.

La presente comunicación con el fin de manifestar mi conocimiento y aprobación del trabajo de grado titulado "ANÁLISIS COMPARATIVO DE DOCUMENTOS FRENTE A OBJETIVOS ESTABLECIDOS USANDO ANALÍTICA DE TEXTO: Caso de estudio Syllabus de los cursos de Unisinú.", elaborada por los estudiantes Eder Ortega Cardona de cédula de ciudadanía C.C No. 73.184.700 y Aldair Cano Yanez de cédula de ciudadanía C.C No. 1.052.737.983, presentado como requisito para optar al título de Ingeniería de Sistemas.

Cordialmente,

Eugenia Luz Arrieta Rodríguez
Coordinadora de Investigaciones de la Escuela de Ingeniería de Sistemas
Universidad del Sinú

Contenido

Lista de Figuras

Lista de Tablas

1 Resumen

El presente proyecto consiste en el proceso de analizar y comparar documentos con objetivos previamente establecidos, con la finalidad de reducir tiempos y costos, y junto a esto determinar la precisión de los temas plasmados en el contenido de los Syllabus de la Escuela de Ingeniería de Sistemas con respecto a los temas que se evalúan en las pruebas Saber Pro y Saber TyT.

El objetivo de dicho proyecto se basa en la obtención de dicho documento en los tipos de formatos WORD, PDF y/o Imagen, para posteriormente llevarlo a un proceso de análisis, y luego de esto comparar y determinar si la temática que se encuentra plasmada en los Syllabus de la Escuela de Ingeniería de Sistemas en sus distintos semestres, corresponde a la que se está evaluando en las pruebas realizadas por el Estado conocidas como las Saber Pro y las Saber TyT. Actualmente esta evaluación se basa en la taxonomía de Bloom [1].

Usando como herramienta la minería de textos, que se encarga de descubrir la información que no se encuentra de forma explícita en ningún texto, pero da surgimiento al relacionar el contenido de varios de ellos. Cabe destacar que esta se encuentra comprendida en algunas fases importantes para su desarrollo, las cuales vienen siendo la recuperación de la información, la extracción de la información y la minería de datos, esta última para encontrar asociaciones entre los datos claves que previamente fueron extraídos. Estas fases a su vez deben ser divididas en tres etapas, que vendrían siendo la de pre-procesamiento, que es la que transforma en algún tipo de representación estructurada o semiestructurada que facilite posteriormente el análisis. Luego viene la etapa de descubrimiento, en la cual se analizan las representaciones internas con el fin de obtener nueva información. Y por último la etapa de visualización, que es donde los usuarios pueden observar y explorar los resultados que fueron obtenidos. Al final se obtuvo una aplicación para gestionar los syllabus. Permite la carga de documentos, se divide en secciones y se realiza un análisis a cada sección. Esto se muestra en forma de informe gráfico. Este muestra, por cada sección, la presencia de los niveles de la taxonomía de Bloom. La forma en que se determina la existencia o no de un nivel, se hace mediante la revisión de las oraciones dentro de la sección. Es decir, las oraciones se analizan de forma individual y se genera un indicador numérico que se muestra dentro del informe.

1.0.1. Abstract

The present project consists of the process of analyzing and comparing documents with before established aims, with the purpose of reducing times and costs, and close to this to determine the precision of the topics formed of the content of the syllabus of the school of systems engineering with regard to the topics that are evaluated in the tests To know Pro and To know TyT.
The aim of the above mentioned project bases on the obtaining of the above mentioned document on the types of formats WORD, PDF and / or Image, later to take it to a process of analysis, and after this to compare and to determine if the subject matter that is formed in the syllabus of the school of systems engineering in his different semesters, corresponds the one that is evaluated in the tests realized by the condition known like To know Pro and To know TyT. Actually, this assesment is based on Bloom's taxonomy [1].
Using as tool the mining industry of texts, which is the manager of discovering the information that one was not finding so explicitly in any text, but it gives emergence on having related the content of several of them. It is necessary to emphasize that this one is understood in some important phases for his development, which come being the recovery of the information, the extraction of the information and the data mining, the latter to find associations between the key information that before were extracted. These phases in turn must be divided in three stages, which would come being that of pre-processing, which is the one that it transforms into some type of structured or semistructured representation that facilitates later the analysis. Then there would come the stage of discovery, in which the internal representations are analyzed in order to obtain new information. And finally the stage of visualization, which is where the users can observe and explore the results that were obtained.
At the end the project end up with an application for syllabus's management. Allowing the load of documents, secction's document split and a section analisys. This shows, on each section, if there is a Bloom's taxonomy level. The way of noticing the existence or not of a level is by checking the sentences on each section. That means each sentence is individually cheched and this generate a numeric indicator on a report.

2 Introducción

El origen de la computación está ligada, indudablemente, a los datos. Quizás se podría decir que el procesamiento de estos da origen a las computadoras. Sustentado por el estudios de varios matemáticos que impulsaron su diseño y creación. La forma inicial de los datos era bastante sencilla y su volumen, pequeño en comparación con el almacenamiento actual. En cuanto al aspecto físico se ha pasado por fichas, engranajes, tarjetas perforadas hasta los bits en forma magnética o electrónica. Es decir se pasó de la mecánica a la electrónica y finalmente al software. Con este último ya se habla de sistemas operativos, sistemas de archivo y, por supuesto, el concepto de archivo digital y directorio. Estos dos últimos nacen de lo que algunas bibliotecas, bancos, instituciones públicas, etc, utilizaban como un conjunto de documentos físicos, relacionados a una persona, caso o tema específico y se guardaban en carpetas de cartón, las cuales eran rotuladas usando una pestaña en su parte superior.
Los archivos se asocian a un tipo y se guardan dentro de los directorios. Esto facilita su ubicación y su organización. Con el tiempo, el uso extendido de las aplicaciones y el aumento de la cantidad de archivos, se cae en cuenta de la necesidad de optimizar su búsqueda y facilitar la relación entre ellos ya que se almacenan como unidades independientes. A partir de aquí nacen los paradigmas que buscan estructurar los datos. La consecuencia lógica de esto son las bases de datos. Esto conlleva a la evolución de las herramientas de diseño de software, incluyendo el concepto de tablas y normalización, el origen del lenguaje SQL (Structured Query Language) y el desarrollo de los motores. Estos últimos se encargan de gestionar la información, mantener la integridad, controlar la seguridad y proporcionar gran desempeño.
Los motores de base de datos tienen un uso muy extendido y la inteligencia artificial no está exenta. Desde sus inicios, la IA (Inteligencia Artificial) trata de modelar, mediante una abstracción, la inteligencia humana. Su evolución da origen al concepto de Machine Learning. Este tiene como objetivo predecir un resultado o comportamiento, incluye un conjunto de valores de entrenamiento como su insumo base, entrena y finalmente aplica una fórmula, por lo general estadística, que le permite ejecutar un proceso de inferencia. Teniendo esto en cuenta, la persistencia, tanto para los datos de entrenamiento, como los resultados, deben apoyarse en una base de datos. Esto garantiza eficiencia en el acceso y gestión de la información, sin preocuparse de su volumen.
El crecimiento del volumen de información estructurada, supone cierto grado de especialización y evolución en el área de las bases de datos. Esto resulta en el nacimiento de los conceptos de Big data, Datawarehouse o Almacén de datos y minería de datos. En esencia todos buscan lograr un objetivo común, definir la estructura mas eficiente para la persistencia de los datos y precisar un método óptimo para extraer información. Descubrir conocimiento sobre datos estructurados

se ha convertido en toda una ciencia. Se denomina, más concretamente, minería de datos. Esta se apoya en múltiples técnicas matemáticas para la extracción de conocimiento sobre grandes volúmenes de datos.
La minería de texto involucra dos disciplinas muy importantes actualmente, incluso se podría decir que es una convergencia de ambas. Una de ellas es la minería de datos y la otra es el procesamiento de lenguaje natural (NLP). La primera se enfoca en el manejo de grandes volúmenes de datos estructurados. Aporta las características de extracción de información. La segunda es uno de los campos mas influyentes de la inteligencia artificial. Actualmente es uno de los de mayor desarrollo y sus aplicaciones son ampliamente extendidas en todo el mundo. Trata de modelar el entendimiento del lenguaje natural usando las computadoras.
Todas estas técnicas se pueden aplicar a diferentes campos. En cada una de ellas se busca solucionar una problemática específica. Algunos de estos no buscan descubrir conocimiento, sino más bien resumir para facilitar su análisis. El presente proyecto tiene este enfoque, busca solucionar el problema comparando documentos contra objetivos de contenido dados. Estos últimos son establecidos por el usuario según sus criterios. Dado esto, se analiza que tanto cumple el documento con las metas propuestas. Los resultados finales van a permitir la toma de decisiones. Las cuales van a incidir de manera directa en el contenido del documento, incluyendo su estructura.
Para llevar a cabo lo descrito anteriormente, el propósito es desarrollar una herramienta software que facilite esta tarea. Su especificación ha quedado plasmada en el presente documento, pero antes que nada se ha definido claramente cual es el problema, se ha realizado una investigación preliminar y se ha determinado el marco de referencia. Teniendo esto se arma la hoja de ruta y la metodología de trabajo. A continuación se describe con mas detalle el trabajo a realizar, abordando la revisión bibliográfica e incluso se describe el modelo de desarrollo de software a utilizar.

3 Diseño Metodológico

3.1. Planteamiento del problema

3.1.1. Descripción del problema

Las pruebas Saber Pro y Saber TyT [2] hacen parte de una estrategia nacional para evaluar las competencia de profesionales, tecnólogos y técnicos que se gradúan en el país. No está demás, decir que son de gran importancia. Actualmente, incluso, constituye uno de los requisitos para poder obtener el título de grado. Su objetivo es medir las competencias de los graduandos. Cada competencia se agrupa en uno de dos conjuntos, el de generales o el de específicas. Al evaluar estos dos grupos se obtiene un indicador de calidad de la educación superior. Con este también se puede ponderar el nivel de las Universidades Nacionales. Finalmente se puede hacer una comparativa entre los programas académicos ofrecidos.
Conociendo que estas pruebas son de carácter obligatorio y la importancia del indicador que representan, la Universidad del Sinú se dio a la labor de revisar el resultado de sus alumnos. Los puntajes obtenidos no eran muy satisfactorios. Debido a esto se dieron a la tarea de investigar las causas y tomar las medidas correctivas necesarias. Una de estas se enfoca en el análisis de los contenidos del Syllabus de cada curso. La Escuela de Ingeniería de Sistemas, de la Universidad del Sinú, vio aquí una oportunidad para hacer un aporte a la solución de esta problemática.
Teniendo en cuenta lo anterior, la Escuela de Ingeniería de Sistemas, desea validar el contenido del Syllabus de sus cursos. El objetivo es analizar si el contenido cumple con las competencias que evalúa las pruebas Saber Pro y las Saber TyT. La validación se puede realizar mediante minería del texto. Para este caso específico se busca que el documento posea un grupo específico de palabras. Estas últimas se agrupan dentro de los niveles de la taxonomía Bloom. Es decir, que se busca clasificar el contenido del syllabus dentro de los niveles de la taxonomía mencionada. Este último se medirá en un porcentaje de coincidencia, según lo dispuesto por las Saber Pro y las Saber TyT. El porcentaje puede ser global o específico dentro de las secciones del Syllabus. Actualmente solo sería posible realizar esta tarea de forma manual.
Dada la cantidad de cursos, esto supone un gran volumen de información. La mejor forma de dar una buena solución es automatizar el proceso, pero actualmente no existe el medio para hacerlo. De forma resumida el proceso de análisis posee cuatro pasos:

- Carga y preparación del documento, el cual puede estar dado en diferentes formatos.
- Definición el grupo de objetivos para comparar, es decir, determinar los parámetros para

validar el grado de coincidencia con las competencia evaluadas en las Saber Pro y Saber TyT.

- Realizar el proceso de análisis teniendo en cuenta un grupo de objetivos propuesto.
- Reportar los resultados mostrando el grado de coincidencia con los objetivos dados.

3.1.2. Formulación del problema

¿Cómo se puede diseñar y desarrollar un sistema de información con técnicas de minería de texto, para la revisión del contenido de los Syllabus de la Escuela de Ingeniería de Sistemas, de manera que se pueda detectar que niveles de la taxonomía de Bloom [1] están presentes, y comparar los resultados contra las competencias evaluadas en las pruebas Saber Pro y Saber TyT?

3.1.3. Justificación

Como se ha descrito en el punto anterior, este proyecto se basa en una problemática específica. Hace parte de un proyecto de investigación mas grande. Es importante porque fomenta la práctica de la investigación en la Universidad y deja una buena experiencia en quienes lo desarrollan. Su pertinencia es totalmente válida, debido a que ataca una dificultad claramente identificada con respecto a las pruebas Saber Pro y Saber TyT. Hay que tener en cuenta que estas pruebas son un indicador de calidad a nivel nacional. Desarrollar un sistema que aporte en el mejoramiento de sus resultados es de gran valor para la institución.

Dado el enfoque de investigación con que se aborda este proyecto y el tema de las tecnologías de minería de textos con el que se desarrolla la solución, los resultados finales pueden dar un aporte en conocimiento, incluso pueden servir de base para futuras investigaciones o nuevos productos. Ligado a esto, está el hecho de que el producto final es una herramienta software. Esta sumará un grano de arena en la mejora del problema planteado. La Universidad lo usará en un contexto real. Esto quiere decir finalmente, que se hará un aporte en tecnología en forma de un aplicativo software.

Con todo lo anterior lo más importante es la solución al problema. Que es el objetivo principal del presente proyecto. Se busca analizar el contenido de los Syllabus de los cursos de la Escuela de Ingeniería de Sistemas y demás programas. El producto que se desarrollará automatizará este proceso, apoyándose en técnicas de minería de textos. Los resultados del análisis se compararán contra las competencias que se evalúan en las pruebas Saber Pro y Saber TyT. Esto permitirá realizar los ajustes necesarios a los contenidos de los Syllabus, incluso mejorar su estructura, es decir, la forma como está organizado y las secciones que contiene. Finalmente la herramienta cumplirá con los requisitos necesarios para lograr lo explicado anteriormente.

La solución podría extenderse al ámbito social ya que, no solamente sería útil para la Universidad del Sinú, sino que también puede ser aplicado en cualquier institución educativa de educación

superior. Esto se soporta en que la problemática expuesta puede ser una realidad en otras instituciones. Lo que demuestra un aporte significativo, debido a que se mejora la calidad de los programas, llevándolos a cumplir con los contenidos que permiten evaluación de competencias en las Saber Pro y Saber TyT.

3.1.4. Objetivos

Desarrollar un sistema de información de análisis técnico de documentos, para identificar el nivel de coincidencia de estos con respecto a un conjunto de requisitos dados, usando técnicas de minería de texto.
Para cumplir con este objetivo se plantean los siguientes objetivos específicos:

- Realizar el acopio de información para determinar los requerimientos funcionales y no funcionales del sistema, mediante reuniones y entrevistas con los interesados.
- Determinar los requisitos de tecnología para cumplir los requerimientos del sistema, mediante la realización de una investigación.
- Diseñar la aplicación conforme a los requerimientos obtenidos para delimitar las tareas de desarrollo, usando UML para la creación de los diagramas.
- Desarrollar un componente de análisis texto, para realizar minería y clasificación de contenido conforme a un grupo de requisitos, de manera que se pueda establecer cuales de ellos están presentes en un documento dado.
- Desarrollar la aplicación que permita dar solución a la problemática expuesta, mediante la implementación del diseño realizado.
- Evaluar el producto obtenido para determinar el cumplimiento de los requerimientos, mediante el uso de técnicas de prueba de calidad de software.

3.2. Estado del arte

A continuación se detallan de forma ordenada, los trabajos realizados por diferentes autores investigados dentro del ámbito del tema del presente trabajo, se demarca su referencia, se describe su objetivo y se detallan los resultados obtenidos. De igual manera se ha planteado un conjunto de temas con el fin de clasificar cada trabajo y facilitar la orientación acerca de la temática que cada uno trata. Los temas son: aplicaciones prácticas, que se refiere a las variadas formas en las que se puede aplicar la minería de textos, en este se describen tres trabajos, incluyendo uno Colombiano. El segundo es herramientas, que se refiere a librerías, aplicativos y lenguajes de programación, en este se incluyen dos trabajos. Finalmente, nuevos paradigmas, que se refieren a nuevos conceptos propuestos en el área de estudio en cuestión. Cabe resaltar que los trabajos no tienen una antigüedad mayor a siete años.

3.2.1. Aplicaciones Prácticas

En México, en la Universidad Autónoma de Nuevo León, el trabajo de tesis, *Cuantificación del interés de un usuario en un tema mediante minería de texto y análisis de sentimiento* [3], tiene como objetivo describir el proceso de cuantificación de interés de un usuario de Twitter en un tema y el análisis de sentimiento de sus comentarios sobre este. Este trabajo presenta una guía de los resultados de un caso de estudio en minería de texto. El cual implementó un sistema que busca identificar, dentro de los comentarios de un usuario en Twitter, un conjunto de temas de interés, ponderando las palabras mediante el indicador TF-IDF. Finalmente se desarrolla una aplicación, totalmente en español, para la clasificación de sentimientos, llamada TOM (Twitter Opinion Mining). Esta herramienta identifica si un comentario es positivo, negativo o neutro, usando una base de datos de palabras, en español, clasificadas de la misma manera.

En Ecuador, en la Universidad de las Fuerzas Armadas - ESPE, el artículo científico *Determinación de niveles de agresividad en comentarios de la red social Facebook por medio de Minería de Texto* [4], tiene como objetivo la clasificación de las personas o entes cibernéticos de acuerdo a su nivel potencial de amenaza para el resto de la comunidad virtual. Los resultados son: Utilizando alternativas de minería de textos para analizar los comentarios realizados por los usuarios y, estos a su vez, interpretados por diccionarios que permiten saber si existen palabras ofensivas y por medio de un algoritmo conocido como Distancia de Levenshtein que determina una medida de "similitud" o "cercanía" entre dos palabras o cadenas de caracteres, se procede a la clasificación de los niveles de agresividad de dichos comentarios; ya sean estos bajos, medios y/o altos.

En Colombia, en la Pontificia Universidad Javeriana, el trabajo de tesis, *Diseño de una metodología para la extracción de funciones y mapas relacionales a partir de herramientas de minería de texto* [5], tiene como objetivo definir un método para extraer, a partir de correos electrónicos, el perfil del cargo, incluyendo responsabilidades y relaciones con otras áreas, de los empleados con cargo de trabajadores del conocimiento, utilizando minería de texto. Los resultados son: Un algoritmo bien definido que busca, usando minería de texto de forma guiada dentro de un conjunto de correos monitoreados para ciertos empleados, extraer la información que permite armar el perfil de dicho empleado, descubrir sus funciones y las relaciones que pueden tener con otros grupos de trabajadores de otras áreas.

3.2.2. Herramientas

En España en la Universidade de Vigo, el trabajo de tesis, *Text Analytics para procesado semántico* [6], tiene como objetivo realizar un análisis de las diferentes técnicas de minería de texto y realizar una implementación práctica que permita dar a conocer su funcionamiento y eficiencia, enfocado principalmente en el análisis semántico. Los resultados obtenidos son: Dentro del desarrollo de la tesis, los ejemplos y explicaciones, se han desarrollado usando el lenguaje R. Las técnicas de minería de texto se aplicaron con la librería tm de R. El resultado final es la aplicación Shiny

[7]. Actualmente está disponible para su descarga y uso con R. Shiny permite el desarrollo de un ambiente Web para navegar, de manera interactiva, sobre los resultados del análisis de la minería sobre un documento. Cuenta con gráfico de barras para los palabras más frecuentes, nube de palabras, tabla con los datos de la matriz documentos-términos, entre otros. Actualmente hace parte de las librerías más relevantes de R.

La editorial Packt Publishing publicó el libro, *Python Natural Language Processing* [8], el cual tiene como objetivo describir los fundamentos de NLP (Natural Language Processing), como se puede implementar de forma práctica sobre cualquier documento, usando la librería NLTK en el lenguaje Python y como se puede enriquecer el entendimiento del lenguaje natural aplicando Machine Learning y Deep Learning. Los resultados obtenidos son: En este libro se demuestra, con ejemplos prácticos, que Python y las diferentes librerías de NLP, tales como NLTK, Polyglot, Standford CoreNLP, entre otras, constituyen una herramienta ideal para la implementación de Procesado de Lenguaje Natural. Pasa por la descripción y procesado del corpus, los algoritmos de NLP y la Ingeniería de Características, la aplicación de Machine Learning y Deep Learning en NLP y otras herramientas avanzadas.

En Estados Unidos, la patente, *System and method for using text analytics to identify a set of related documents from a source document*, con número US 9495349 B2 [9], tiene como objetivo describir un sistema que analiza un documento fuente, extrae información relevante y la estructura de tal forma que se puede buscar un conjunto de documentos relacionados. Los resultados son: Antes que nada el sistema requiere una base de datos con ciertas características. Contiene una estructura de meta-datos que permite indexar un conjunto de documentos. Teniendo esto definido, el sistema queda compuesto de cuatro componentes: El sistema de analítica de texto, el sistema de comparación, el sistema de ranking y agregación y finalmente el de anotación. El primero realiza el análisis y la extracción de información del documento fuente, organiza los datos extraídos y arma una estructura que posteriormente se usará para la búsqueda de documentos relacionados. El segundo optimiza la consulta a la base de datos, toma como insumo la estructura del sistema anterior y la compara con los meta-datos para identificar los textos semejantes. El tercero realiza un ranking a los resultados anteriores y ayuda a afinar su clasificación. Finalmente, el último sistema, realiza las anotaciones pertinentes a la estructura hallada en el documento fuente.

3.2.3. Nuevos paradigmas

En España en la Universidad de Granada, el trabajo de tesis, *Nuevas técnicas de minería de texto: Aplicaciones* [10], tiene como objetivo describir los antecedentes que dan origen a la minería de texto, detallar los pasos involucrados en su proceso, profundizando en las unidades que lo componen, para finalmente explicar un nuevo paradigma basado en Conocimiento y una aplicación de búsqueda de contradicciones. Los resultados son: En este trabajo se formula un

nuevo paradigma de minería de texto basado en Conocimiento. Habitualmente el descubrimiento de conocimiento mediante la minería se hace de forma inductiva, pero en este nuevo paradigma se propone que sea deductiva. Se basa en el estudio del funcionamiento de los sistemas expertos. Como es sabido, los agentes inteligentes trabajan con una base de conocimiento, la cual es estructurada y se compone de un conjunto de premisas lógicas bien definidas. Teniendo esto en cuenta, dicho agente predice su funcionamiento estudiando cada premisa que conoce. Este paradigma busca guiar, mediante un objetivo basado en razones, la búsqueda de conocimiento. Compilando premisas sencillas, matemáticamente representables y abordando la técnica desde la perspectiva de los sistemas expertos, se da total validez a este nuevo paradigma.

3.3. Marcos de referencia

3.3.1. Marco teórico

En el mundo del almacenamiento de datos encontramos información estructurada y no estructurada. De la primera, hablamos mas concretamente de bases de datos. De la segunda, cualquier grupo de documentos, incluso los no relacionados. Las bases de datos permiten extraer información fácilmente. Podemos usar lenguajes como SQL para sintetizar lo que queremos buscar. Para la información no estructurada la historia es diferente. Por ejemplo a finales de los años 80 que empieza a resaltar la minería de texto [10]. A partir de aquí se empiezan a gestar las técnicas orientadas a extraer información.

Del lado de información estructurada está el Big data y su consecuente, minería de datos. El procesamiento de grandes volúmenes de datos se apoya en las matemáticas y estadísticas. La información no estructurada se trata de abordar de forma similar, pero se enfoca en NLP (Natural Language Processing) [8]. Es decir, se apoyan en el entendimiento del lenguaje natural y en la automatización del procesamiento de la información. Por entendimiento del lenguaje tenemos la sintaxis, la morfología, los tipos de palabras, entre otros. La automatización busca extraer las palabras, clasificarlas, ponderarlas e incluso encontrar las relaciones entre ellas. Esto es la base para preparar la extracción de la información.

Teniendo en cuenta lo anterior aparece en escena el KDT, por sus siglas en inglés o descubrimiento del conocimiento. Este concepto encierra todo lo concerniente al análisis de documentos. Como lo indica su nombre pretende identificar información, en muchos casos no explícita, dentro de un documento o texto. Está pensado para el procesamiento de grandes volúmenes de documentos. Su objetivo es extraer lo importante usando el conjunto de técnicas adecuadas. La extracción puede ser orientada a un objetivo, es decir, dirigida. También puede ser abierta o sin objetivos aparentes. Para este caso el descubrimiento es mas autónomo y las técnicas un poco diferentes.

Para la implementación de la minería de texto existen varias herramientas. Para el desarrollo, mas concretamente, se destacan dos lenguajes de programación, R y Python. Ambos cuentan con librerías maduras, completas y fáciles de programar. Son lenguajes muy productivos y optimizan el prototipado. La elección de uno u otro es cuestión de gusto.

3.3.2. KDT - Descubrimiento de conocimiento en Texto

Knowledge Discovery in Text o KDT, busca, como su nombre lo indica, descubrir conocimiento. Si se piensa en la gestión documental en las empresas, los correos electrónicos, blogs, sitios web, entre otros, nos damos cuenta que tenemos un gran volumen de información, difícil de catalogar. La información estructurada, como bases de datos, incluso tablas de Excel, tienen relaciones definidas. No es necesario descubrirlas. A partir de aquí preguntar por cierta información, sintetizar, agrupar o relacionar datos, es una tarea que se facilita dada la estructura de datos.
Cuando los datos son textuales, no tienen una estructura aparente, no están explícitamente relacionados y tienen gran volumen, se recurre al KDT para procesarlos. Este inicia un proceso para reconocer los conceptos o temas relevantes en un conjunto de documentos. Extrae la información realmente importante. Encuentra patrones que dan un significado relevante a uno o varios documentos. Si se hace una pregunta puntual, es posible iniciar un proceso de búsqueda y hallar la respuesta. Con KDT se puede descubrir y clasificar información de la que no eramos consciente que estaba ahí. Es decir, es capaz de generar nuevo conocimiento.
Teniendo en cuenta lo anterior, esta técnica se ha convertido en un área de conocimiento relevante para otras disciplinas. Entre las que podemos destacar está el derecho y la medicina. Ambas pueden generar grandes cantidades de información textual. En la primera tenemos por ejemplo derechos de petición, tutelas, demandas, entre muchos otros. Al ser almacenados se clasifican por temas mayores, pero sigue conteniendo un volumen considerado de texto sin estructura. Para el caso de la medicina, existen las historias clínicas de pacientes. Estas pueden estar semiestructuradas, pero pueden crecer exponencialmente dependiendo de la cantidad de pacientes y de la cantidad de consultas. Estas y otras áreas pueden usar KDT para extraer información. Como resumen, KDT se puede utilizar para:

- Descubrir y extraer conocimiento.
- Categorización de temas (En documentos almacenados, sitios web, blogs, etc).
- Minería de opinión.
- Clustering de documentos.
- Análisis guiado de documentos.

KDT también se conoce como Analítica de Texto y está inmerso en la minería de datos [11]. Cubre el concepto de Minería de Texto y las técnicas de preparación que le sirven de insumo. Relaciona el procesamiento de lenguaje natural y la minería de datos como se muestra en la Figura **3-1**. Su funcionamiento se resume en tres pasos: Pre-procesamiento o preparación del documento, procesado o minería de texto y descubrimiento o presentación de los datos encontrados.

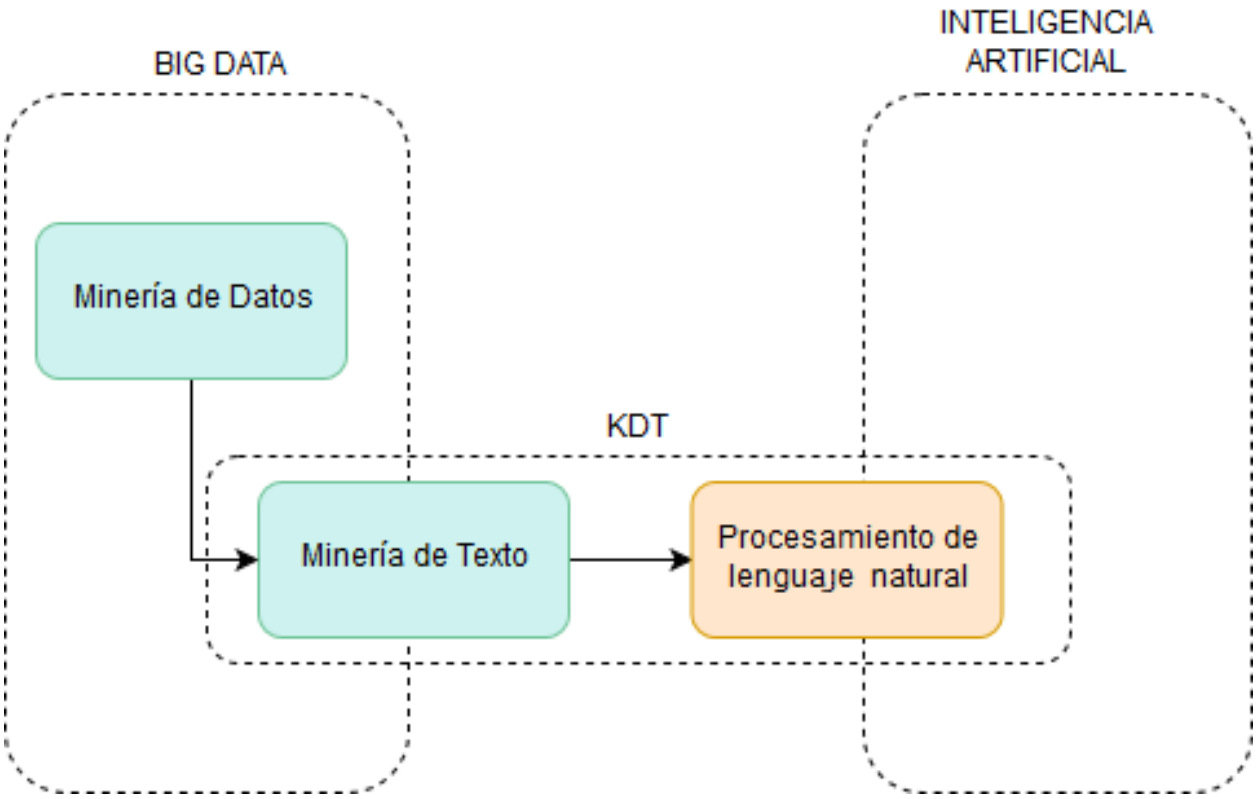

Figura 3-1: Diagrama KDT

Preparación del documento

El insumo inicial para este proceso es el corpus, que es una forma básica de texto, no tiene ningún tipo de formato y es entendible en múltiples plataformas. La preparación del documente busca, inicialmente, disminuir el volumen de la información. Para lograr esto se aplican varias técnicas de forma secuencial. Cada una de estas actualiza el estado actual del corpus, lo mejora paso a paso dependiendo de lo que se busca y facilita la extracción de los datos.
Estas técnicas están inmersas dentro del procesado de lenguaje natural - NLP. Lo principal aquí es la clasificación e identificación de palabras, el manejo de los signos de puntuación, de los números, de símbolos, los nombres de entidades, así como también la lematización para tratar la morfología de las palabras y unificarlas. Paulatinamente, la aplicación de cada método, va disminuyendo la cantidad de datos. A este punto el corpus es mas compacto y hasta se puede decir que está semi-estructurado. Esto mejora notablemente la aplicación de la minería de texto.
Antes de hablar de minería hay que tener en cuenta que, dependiendo de los métodos de preparación aplicados, los resultados del procesado de texto al final pueden variar. Hay que decir que todas las técnicas no son necesarias, porque dependen del tipo de documento y del resultado que se busca. Dependiendo de como se apliquen pueden facilitar o entorpecer la extracción de datos. Es recomendable seguir lo que otros autores han usado con buenos resultados y valerse de su

experiencia. En resumen los procedimientos de esta fase son los siguientes [6]:

- Identificar palabras.
- Convertir a minúsculas.
- Eliminar stopwords.
- Eliminar signos de puntuación.
- Eliminar números.
- Manejar los espacios.
- Agrupar sinónimos.
- Simplificar y etiquetar términos.
- Lematización.

Descubrimiento

Es la ejecución de la minería propiamente dicho. Toma el resultado dejado por la preparación anterior e inicia la extracción de información. Si el documento se preparó de forma adecuada, se obtendrán buenos resultados en esta fase, aunque pueden variar según la preparación. Existen dos enfoques de partida. En el primero se tiene un conocimiento base, es decir se ejecuta de forma dirigida a partir de un objetivo. En la segunda, el propósito no está claramente definido, incluso el producto final es totalmente desconocido. Se puede decir que es descubrimiento en todo el sentido de la palabra.
Si se fija un objetivo claro y se tiene algo de conocimiento del documento a analizar, la minería será mas eficiente. El propósito guía el resultado. Se escogen las técnicas mas adecuadas según el caso. Cada una de ellas trabaja bajo la directriz previamente trazada. Finalmente se puede cotejar el producto obtenido con la meta planteada. Ahora bien, no siempre se tiene un conocimiento claro de los documentos a minar. De igual forma los métodos de esta fase están enfocados en descubrir información.
Dentro de las técnicas mas conocidas tenemos, el resumen, que se puede representar con la matriz de documentos-términos. Busca reducir el documento a un conjunto de palabras. Cada una de estas es ponderada según su grado de importancia. También está la detección de tópicos importantes mediante la definición de términos claves, la relación de conceptos que descubre la semántica del texto, la clasificación de temas, el clustering aplicado a conjuntos de documentos, la respuesta a preguntas y el Deep Learning.

Presentación

Constituye el paso final del proceso de KDT. Es el encargado de visualizar los resultados obtenidos. Dependiendo del producto final obtenido, así se construirá la visual. Esta se puede presentar en un formato estático o interactivo. Cada uno de ellos se puede representar mediante documentos ofimáticos, archivos PDF, páginas HTML, tanto estáticas como las que permiten interacción del usuario, mapas, entre otros. La representación de estos datos puede ser en tablas (fijas o dinámicas), gráficos, mapas de calor, etc. Se puede usar filtros para navegar los resultados, incluso se puede ir de lo mas general a lo mas detallado. Esta última característica se resume en lo que algunos llaman la minería de texto visual [11], debido a que usa herramientas que permiten ir descubriendo los resultados a medida que nos movemos sobre su espacio.

3.3.3. Minería de texto

Es un derivado directo o especialización de la minería de datos. Trata de compensar la aplicación de sus métodos a los datos no estructurados. Más concretamente a la información contenida en textos, busca descubrir conocimiento valioso que no se muestra explicitamente. Encuentra las relaciones que aparentemente no existen y extrae lo realmente importante. Identifica patrones, jerarquiza y agrupa los resultados y, dependiendo de la cantidad, realiza un cluster al conjunto de documentos según los temas encontrados. Se puede realizar una extracción puntual si se tiene una base de conocimiento previo de los escritos. Cabe notar que hace parte del proceso de KDT. Algunos autores incluso mencionan que son términos equivalentes.

La minería de texto entra en escena en la segunda fase de KDT, este puede trabajar preferiblemente sobre un gran volumen de documentos. Luego de descubrir información en muchos casos desconocida, la organiza, la ordena y estructura para almacenarlo en KOS (Knowledge Organization Structure) [12]. De esta forma queda el registro del trabajo realizado y el resultado puede ser consultado posteriormente. Una vez se tiene una estructura, puede relacionarse mas fácilmente con bases de datos o con otros resultados de minería. Es más, ya que el resultado está archivado, no es necesario procesar nuevamente el texto. Esto no quiere decir que no se pueda hacer minería, desde otra perspectiva, al mismo escrito.

Antes de armar el KOS se pasa por tres componentes que son: recuperar información, procesar información e integrar información. Es una subdivisión general de las tareas de minería de texto. Estas pueden ser aplicadas en diferentes ámbitos. Se puede usar en análisis de patentes, para realizar análisis comparativo con la intención de encontrar similitudes. En sitios web, blogs, entre otros recursos web, para clasificar y agrupar información según temas debido a su gran diversidad. En historias clínicas, para identificar indicadores y datos médicos relevantes con el fin de estructurarlos. En e-mails, para minería de opinión y análisis de sentimientos, con lo que se puede hacer mercadeo y detección de tendencias. En artículos científicos, para conformar clusters acerca de tópicos específicos. Entre muchas otras áreas.

3.3.4. Procesado de Lenguaje Natural (NLP)

NLP, por sus siglas en inglés, es una rama importante de la inteligencia artificial. Una de las que ha tenido mas desarrollo y relevancia actualmente. Aplicaciones como SIRI o Cortana dan fé de ello. A diferencia de la minería de texto, NLP está enfocado en entender el lenguaje natural no en extraer o minar datos. Claro está que la analítica de texto se sirve de ella para identificar la información a extraer. La fuente de esta última puede ser tanto en medio escrito como hablado. Jalaj Thanaki[8] Lo define como la habilidad de tecnologías computacionales y/o lingüísticas para procesar el lenguaje natural como lo hacen los humanos. Automatiza el procesamiento de las reglas gramáticas y de sintaxis sobre una pieza o recurso lingüístico. Con esto busca facilitar la interacción entre los humanos y las computadoras mediante el lenguaje natural.
NLP se fundamenta en la matemática, el álgebra lineal y las estadísticas aplicadas a un lenguaje, aunque tiene en cuenta los medios para el manejo de datos estructurados tales como, el lenguaje SQL, formatos de fácil transporte como JSON y XML, entre otros. Por supuesto también se basa en el entendimiento del lenguaje, el léxico, la morfología de las palabras, la gramática, la estructura de oraciones y su sintaxis. Con las matemáticas se modelan los algoritmos para procesar el lenguaje y describirlo. Con la lingüística computacional se identifica el tipo de gramática. Esta cuenta con técnicas de reconocimiento de términos, incluso usa estructuras como los n-gramas para facilitar la relación entre ellos. Si se conjuga lo anterior con el conocimiento de un idioma, entonces surgirá el significado de la pieza de lenguaje que se desea analizar.
Dicho análisis está condicionado a cierto conjunto de requerimientos. Dependiendo de estos se determina el tipo de insumo o corpus que se necesita y si este debe ser o no procesado, es decir, si se tiene que preparar de cierta manera para maximizar su utilidad y mejorar los resultados a obtener. Tal producto se obtiene de una aplicación específica de NLP. Para el desarrollo de estas aplicaciones se pueden usar diversos lenguajes de programación. Según la experiencia de muchos autores, los más ponderados son Python y R. Esto se debe claramente a su alta productividad, su rápido prototipado y su eficiencia, además cuentan con librerías especializadas en esta área. El campo para desarrollar sistemas con NLP es muy vasto. En resumen NLP se puede aplicar dentro de las siguientes áreas:

- Sistemas de reconocimiento de voz.
- Sistemas de respuesta a preguntas.
- Sistemas de traducción.
- Análisis de sentimientos.
- Resumen de textos.
- Clasificación de tópicos.
- Segmentación de temas.

3.3.5. Marco conceptual

Los conceptos claves que se deben tener en cuenta en la presente investigación son:
Corpus: Es la forma básica, que sirve de insumo para el posterior análisis de texto. Debe ser preprosesado para optimizar los resultados. Se representa como un documento, una bolsa de palabras, frases, entre otros.

Stopwords: Representan el conjunto de palabras más frecuentemente utilizado en textos, de manera que no ejercen ningún tipo de aporte valioso.

Lematización: Es un proceso que busca, mediante el conocimiento de la morfología de las palabras, extraer su raíz y prescindir su parte variable. De esta manera se optimiza el proceso de análisis de documentos.

Matriz de documentos-términos: Es una tabla o matriz de frecuencia de términos. Aquí los términos constituyen las columnas y las filas representan cada uno de los documentos. Las frecuencias de aparición de cada término se almacenan en las celdas, es decir en la intersección entre el término y el documento. A mayor frecuencia, mayor importancia del término.

Bolsa de palabras: Es una forma intermedia. Procede de la aplicación de técnicas de procesamiento al corpus. Representa un conjunto de palabras debidamente tokenizadas. Su orden viene dado por la fuente o corpus dado y contiene las palabras relevantes para la minería y descubrimiento de conocimiento.

TF-IDF: Se trata de una medida de ponderación o el grado de relevancia de los términos dentro de un documento. Es el producto de dos dos medidas, la frecuencia del término y la frecuencia inversa del documento. Esto quiere decir que su valor aumenta con el número de veces que aparece una palabra en el documente, pero es compensada con las veces que aparece en la colección de documentos.

Naive Bayes o clasificador Bayesiano ingenuo: Es un modelo clasificador probabilístico llamado ingenuo, dado que sus variables de predicción o de hipótesis se caracterizan por ser independientes. Además, se basa, principalmente, en el teorema de Bayes, que aunque tiene un fórmula sencilla, ha demostrado ser muy útil y confiable para la Inteligencia artificial.

A continuación, en la Figura **3-2**, se describe el ámbito de conocimientos que comprende la analítica de texto o KDT. Muestra los conceptos de los que se deriva y le dan origen, como se aplica y en que se basan cada uno de sus derivados. Describe las aplicaciones y técnicas de sus dos grandes fundamentos, la minería de texto y el procesado de lenguaje natural y además muestra la forma en como estos dos se relacionan. Finalmente se describe un componente importante, el corpus, que constituye el insumo base del que parte el análisis. Detalla los pasos iniciales de su preparación y se listan las formas en que pude ser representado.

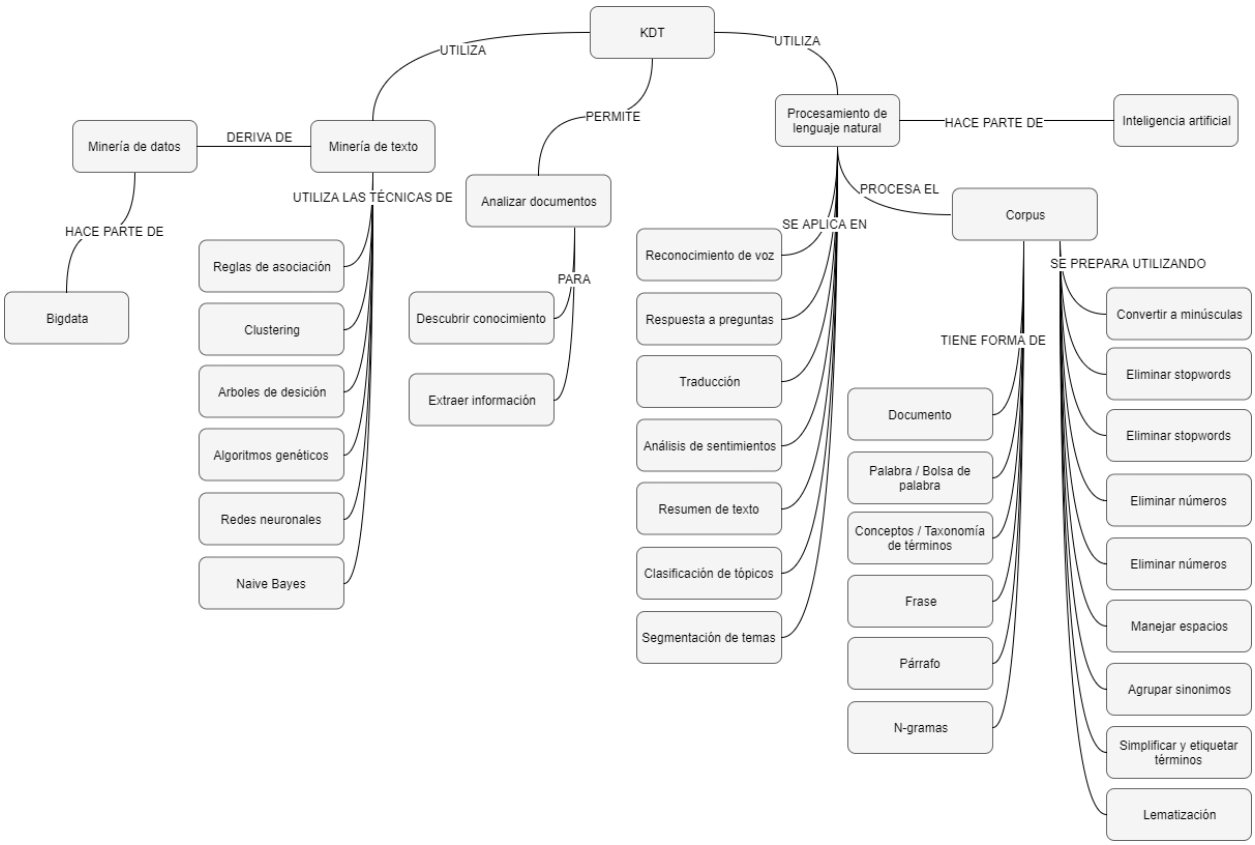

Figura 3-2: Mapa conceptual KDT

3.3.6. Marco legal

3.3.7. Normativa de derechos de autor

Las disposiciones en cuanto a derechos de autor en Colombia se enmarca en dos documentos. El primero es la Ley 23 de 1982 [13], la cual establece las medidas de protección para obras literarias, científicas, artísticas, el producto de los intérpretes, los productores de fonogramas y los organismos de radiodifusión. Para mayor claridad, primero se definen los tipos de productos que cubre la ley y luego se establece el marco de defensa legal para los autores de estos, dentro del territorio nacional de Colombia.

El segundo documento hace una especificación normativa más concreta al tema de software. Se trata del Decreto 1360 de 1989 [14], donde se reglamenta la inscripción de soporte lógico (software) en el Registro Nacional del Derecho de Autor. Donde el software se considera una creación propia del dominio literario. Además, se dan todas las directrices para su inscripción. Con esto y el apoyo de la ley 23 de 1982, los autores, quienes hayan inscrito sus creaciones, pueden proteger sus aplicativos de software.

Finalmente, para aumentar la cobertura de la protección, se creó la Decisión 351 [15], de la comisión del acuerdo de Cartagena, creada por la Comunidad Andina. Con esta, la defensa de

los derechos de autor, se extiende a los países que conforman la Comunidad Andina, Colombia, Perú, Ecuador, Bolivia y Chile.

3.3.8. Licenciamiento de software libre

El licenciamiento permite regular el correcto uso de las copias de software. Determina su forma de acceso, solo ejecutable o código fuente. Es un acuerdo que el usuario final se compromete a cumplir. Existen muchos tipos de licencia que actualmente son estándares, debido a que funcionan perfectamente en ciertos escenarios comunes. Las hay libres (gratuitas y de código abierto) y de propietarios particulares (Solo uso comercial del ejecutable mediante un costo). Por el lado de las libres tenemos la GPL, AGPL, BSD, Apache, entre otras. La que consideramos mas adecuada para el presente proyecto es la GNU-GPL [16].

Cabe notar que la licencia GNU-GPL, no es exclusiva para software. Puede ser aplicado a muchos ámbitos, cualquiera en donde un autor desee poner a disposición su trabajo y que este sea aprovechado y distribuido sin perder su condición "libre". Permite el copyleft con ciertas reglas. Las modificaciones que se hagan al software seguirán siendo libres. Se debe respetar este derecho. Ahora bien, la libertad se trata de la distribución, el acceso al código fuente, el permiso para modificarlo, pero en ningún momento se especifica la restricción de no cobrar por el trabajo. A pesar de lo anterior, se mantienen los derechos de propiedad del autor, aunque la responsabilidad por el funcionamiento de las nuevas modificaciones, no recaen sobre este, de hecho, se deben especificar claramente las nuevas derivaciones del código original.

3.4. Metodología

3.4.1. Línea de investigación

La Universidad del Sinú seccional Cartagena cuenta con varios grupos de investigación, los cuales están distribuidos en sus diferentes programas. Para el caso de la escuela de Ingeniería de Sistemas el grupo se llama DEARTICA. Este cuenta con tres líneas claras de investigación. El presente proyecto hace parte de la línea de Inteligencia artificial. Esto debido a que la minería de texto, activo principal de este trabajo, se apoya en el procesado de lenguaje natural o NLP, la cual se sustenta, para su funcionamiento, en muchas de las técnicas de Machine Learning que se usan hoy día. Dado lo anterior queda clara la participación de la IA.

La Universidad del Sinú fomenta la investigación mediante los grupos antes mencionados y, por su puesto, con sus proyectos. Cabe resaltar que el presente trabajo es un componente de uno de ellos. El cual no solo impulsa las capacidades investigativas, sino que también propende la solución de una problemática particular de la Universidad.

3.4.2. Tipo de investigación

Por la forma en que se está realizando el acopio de la información, la naturaleza de la misma y la aplicación práctica que se le pretende dar, el tipo de investigación usada en este proyecto es la aplicada. Esto se confirma debido a que se creará un nuevo producto a partir de conocimientos ya establecidos. El cual tiene la intención de ser útil dentro del sector educativo, ya que resolverá un asunto importante ligado a contenidos curriculares y su concordancia con lo exigido por las pruebas Saber Pro y las Saber TyT. Puede ser incluso aplicado a otras Universidades nacionales, ya que todas tienen que lidiar el tema de las pruebas.
Teniendo en cuenta lo descrito anteriormente, definimos la investigación aplicada como generadora de conocimiento mediante la aplicación directa a los problemas de la sociedad o el sector productivo [17]. Esto quiere decir que permite llevar a la práctica las teorías y tecnologías desarrolladas mediante la investigación básica. Busca generar un valor agregado creando productos útiles a la sociedad o mejorando las condiciones de ciertos sectores. Esto conlleva naturalmente al progreso de la tecnología.

3.4.3. Definición de la metodología

Para la formulación y desarrollo de esta investigación se decidió trabajar en la metodología de investigación presentada por Vaishnavi y Kuechler [18]. La cual es ideal para proyectos de investigación en tecnología. Es una metodología madura, de estructura lógica y fácil de implementar. Consiste en cinco pasos generales, pero bien definidos:

1. Conciencia del problema, en la cual se conoce la naturaleza de la problemática y se expone la propuesta inicial conforme a los requerimientos del método o artefacto a crear.

2. Sugerencia, define el diseño, ajustado a los requisitos, que permite generar una solución.

3. Desarrollo, es la etapa de ejecución que dará como resultado un método o artefacto siguiendo el diseño propuesto.

4. Evaluación, define y ejecuta las pruebas necesarias que permitan medir la calidad del producto obtenido.

5. Conclusión, en la cual se resumen los resultados obtenidos durante la elaboración de cada uno de los pasos.

Esta metodología permitió, mediante la alineación con los objetivos de este proyecto, crear un modelo de tareas bien definido. Estas actividades, ejecutadas en un orden lógico, han permitido lograr cada objetivo de este proyecto de una forma eficiente. Para esta adaptación cada paso agrupa un conjunto de tareas, mientras que su realización puede llevar a alcanzar una o dos metas.

- En el primer paso se estableció la recolección y validación de los requerimientos mediante entrevistas. Se definió el formato IEEE 830 [19] para aterrizar y estructurar la información. Finalmente se definió la ruta de investigación para determinar los requisitos tecnológicos del aplicativo a desarrollar. Con esto se cubren los objetivos uno y dos del proyecto.
- En el segundo paso se definió el lenguaje UML como la herramienta de diseño. Se escogieron los diagramas necesarios para modelar la aplicación y se definieron las tareas de diseño ajustados a los requisitos establecidos. Con esto se cubre el objetivo numero tres del proyecto.
- En el tercer paso se definió el lenguaje, las librerías, el framework y la base de datos para el manejo de persistencia. Las tareas de desarrollo se dividieron entre desarrollo de análisis y minería de texto y desarrollo de aplicación de gestión y operación. Con esto se cubre los objetivos número cuatro y cinco del proyecto.
- En el cuarto paso se definieron el conjunto de pruebas a realizar y las tareas de validación de los requisitos del producto. Con esto se cubre el objetivo número seis del proyecto.
- En el quinto paso se definieron las tareas de documentación de las actividades realizadas y la formalización del presente documento.

4 Análisis del problema

4.1. Requerimientos del sistema

A continuación se describen los requerimientos del sistema divididos en cuatro componentes. Inicialmente se detalla la Descripción general del sistema, luego se describen las Interfaces Externas, si las hay. Posteriormente se describen las Características del sistema y finalmente se listan los requerimientos no funcionales del sistema. La especificación de requisitos se realizó utilizando el estándar IEEE 830 [19].

4.1.1. Descripción general

1. **Perspectiva del producto:** Provee un mecanismo de medición de coincidencia de los Syllabus con respecto a las competencias que se evalúan en las pruebas Saber Pro y Saber TyT.

2. **Funciones del producto:**

   - Provee un medio para la carga de documentos en diferentes formatos (pdf, word o imagen).
   - Permite establecer un conjunto de objetos de análisis de documentos.
   - Analiza un documento cargado con respecto a los objetivos establecidos.
   - Muestra los resultados del análisis indicando el porcentaje de coincidencia con respecto a los requisitos establecidos.
   - Almacena los resultados del análisis realizados para posteriores consultas.
   - Ofrece un medio para consultar resultados de análisis almacenados.

3. **Clases de usuario y características:**

   - **Usuario de consulta**: puede consultar los resultados almacenados.
   - **Usuario de gestión**: permite configurar y realizar la carga y análisis de documentos.
   - **Usuario administrador**: permite la creación de los usuarios y la asignación de sus permisos de acceso.

4. **Entorno de operación:** El sistema será desarrollado para operar en sistemas Linux. Debido a que es un aplicativo web, funcionará en un servidor Apache o compatible. Accederá a una base de datos PostgreSQL para la gestión de sus datos.
5. **Restricciones en diseño e implementación:**
6. **Documentación de usuario:** Manual de usuario del sistema.
7. **Suposiciones y Dependencias:** Lenguaje Python v2.7, framework Django v1.8.2, librería NLTK, Motor de base de datos PostgreSQL.

4.1.2. Interfaces externas

1. **Interfaces de usuario:** No aplica en este proyecto.
2. **Interfaces de hardware:** No aplica en este proyecto.
3. **Interfaces de software:** No aplica en este proyecto.
4. **Interfaces de comunicación:** No aplica en este proyecto.

4.1.3. Características del sistema

1. **Gestión de requisitos:**
   - Descripción y prioridad: Permite guardar, editar o eliminar los requisitos, incluye la definición de los cursos y los programas, estableciendo los términos necesarios para cada curso, su frecuencia de aparición y el porcentaje dentro de todo el programa.
   - Estímulo/Secuencia de respuesta: Se selecciona el programa y de acuerdo a cada curso, se establece el porcentaje de cada uno y se ingresan los términos necesarios junto con su frecuencia.
   - Requerimientos funcionales:
     - Consulta y selección de programas.
     - Agregar/editar/eliminar requisitos.
2. **Análisis de documentos:**
   - Descripción y prioridad: Apoyándose en una herramienta para analítica de texto y, tomando como entrada los requisitos dados por el usuario, procede a realizar un análisis que permita determinar el nivel de coincidencia del documento con respecto a los requisitos dados. Finalmente almacena los resultados para posteriores consultas.
   - Estímulo/Secuencia de respuesta: Se selecciona el requisito, se carga el documento a analizar, se ejecuta el análisis.

- Requerimientos funcionales:
  - Consulta y selección de requisitos.
  - Analizar documentos.
  - Guardar resultados de análisis.

3. **Consulta de resultados:**
   - Descripción y prioridad: Cuenta con varias opciones de filtrado y consulta, con lo cual se puede encontrar la información deseada. El filtro puede ser por fecha, código de documento o programa. Los resultados se mostrarán en una tabla de datos, incluyendo la opción de visualizar los datos.
   - Estímulo/Secuencia de respuesta: Se seleccionan las opciones de filtrado, se ejecuta la consulta y, de manera opcional, se pueden visualizar los datos a manera de informe.
   - Requerimientos funcionales:
     - Opciones de selección para filtrado de consulta (Fecha, código documento o programa).
     - Mostrar tabla de resultados de consulta y opción de visualización mediante informe.

4. **Visualización de resultados - Informe:**
   - Descripción y prioridad: Toma la información de resultado de un análisis, ya sea por consulta o durante el proceso de creación. En cuanto a la visualización de resultados, se mostrará una tabla con el resumen de la coincidencia encontrada y se apoyará en gráficos porcentuales para facilitar su interpretación.
   - Estímulo/Secuencia de respuesta: Se ejecuta la visualización y, de manera opcional, se exporta el informe resultante.
   - Requerimientos funcionales:
     - Recibir como entrada los resultados de un análisis realizado.
     - Mostrar una tabla con el resumen de las coincidencias y una gráfica porcentual de respaldo.
     - Habilitar una opción para exportar en PDF el informe actual.

4.1.4. Otros requerimientos no funcionales

1. **Requerimientos de desempeño:** La función de análisis de documentos debe ser de alto rendimiento, de manera que se evite el bloqueo de la aplicación o la espera injustificada por parte del usuario. Debido a que será una aplicación Web, se espera que el acceso a la misma sea eficiente dentro de la Intranet de la institución.

2. **Requerimientos de protección:** No aplica en este proyecto.

3. **Requerimientos de seguridad:** El acceso al aplicativo estará controlado por un sistema de autenticación de usuario. La institución debe establecer las políticas de protección de nombres de usuario y contraseñas para los usuarios del sistema.

4. **Atributos de calidad del software:** El sistema debe cumplir con la totalidad de los requisitos funcionales. Además, las técnicas de minería de texto, deben poseer un alto grado de exactitud.

5. **Reglas del negocio:** El sistema controlará el acceso de solo para los usuarios registrados, las características de acceso a las funciones del mismo estarán limitadas a los permisos de cada tipo de usuario. El usuario administrador tendrá la facultad de crear nuevos usuarios y asignarle su tipo de permiso. Además, si así lo exige la circunferencia, el usuario administrador podrá desactivar una cuenta de usuario.

   Adicional a esto, debe ser una aplicación Web, incluyendo una interfaz intuitiva y de fácil uso.

5 Diseño de la solución

5.1. Casos de uso

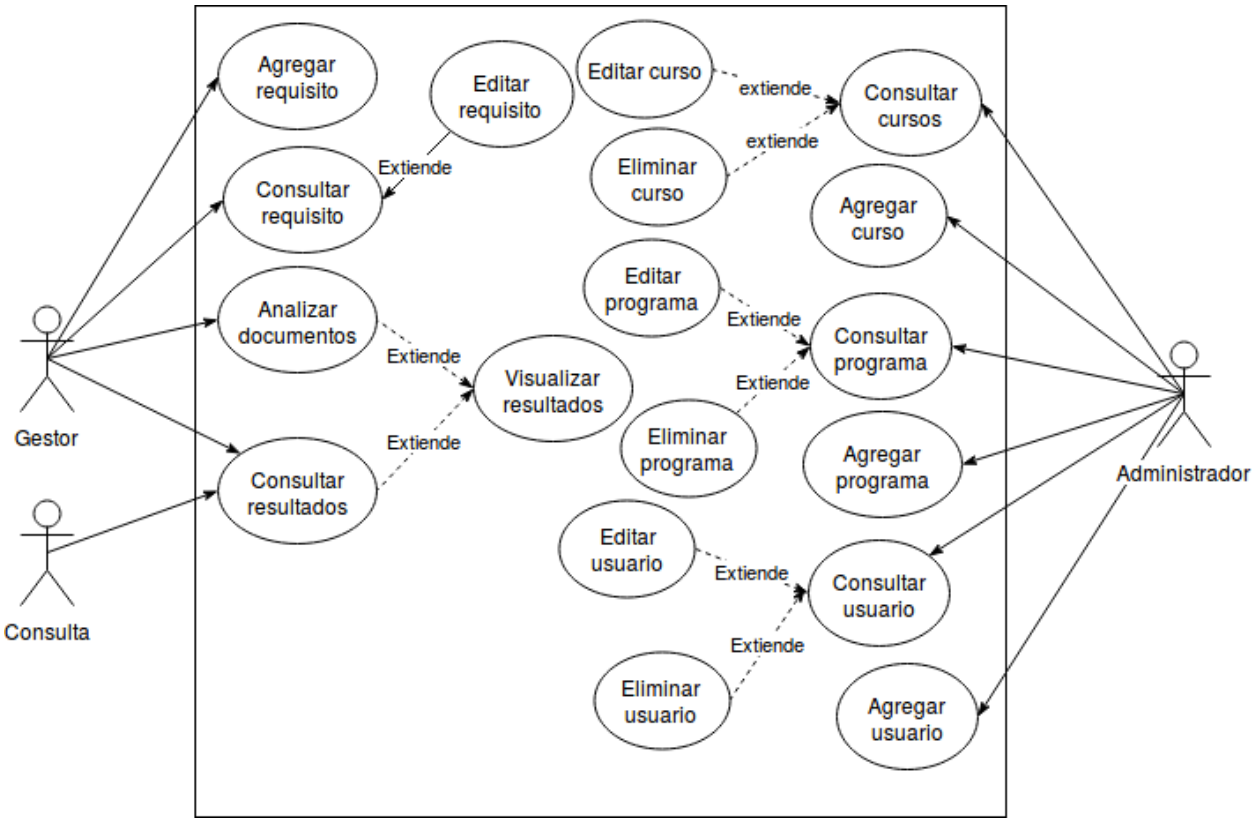

Figura 5-1: Casos de uso

En la actualidad para la creación o implementación de cualquier tipo de software con todas las especificaciones que este requiere, es necesario que se lleve a cabo el cumplimiento de algunos requerimientos previos para realizar un buen desarrollo de éste. Entre estos se encuentran los casos de uso, que son utilizados para especificar la funcionalidad cuando se trata de sistemas que tienen un alto grado de interacción entre el hombre y la máquina. Como bien se dijo, estos son muy útiles para capturar requerimientos, permiten ayudar con la definición de la arquitectura y a su vez establecer pautas para el diseño y las pruebas que deben realizarse. En la figura **5-1** se describen los casos de uso con cada una de las actividades que realiza, cada uno de los

actores que está involucrado en el uso o implementación de dicho software. Se muestran cada una de las acciones que puede realizar un determinado actor basándose en los requerimientos que fueron previamente establecidos para poder llevar a cabo el desarrollo del software en mención. Posteriormente estos casos deben ser especificados, es decir hacer la descripción de cada una de las partes o acciones definidas para lograr la descripción completa y detallada, basándose en un formato específico. Estas descripciones de cada caso se pueden ver en las Tablas que van de la 5-1 hasta la 5-18.

Tabla 5-1: Caso de uso - Agregar curso

No. Caso	1001	**Prioridad**	Normal
Nombre	Agregar cursos		
Actor	Administrador		
Objetivo	Guardar los datos de los cursos		
Pre-condición			
Flujo ideal	Actor: 2) El administrador rellena el formulario con los datos solicitados. 3) El administrador selecciona la opción guardar los datos.	Sistema: 1) El sistema despliega una ventana con el formulario de registro. 4) El sistema almacena en la base de datos, la información ingresada por el administrador. 5) El sistema muestra una ventana de confirmación de los datos almacenados.	
Flujo alterno	Actor: 3.1) El administrador tiene campos vacíos y/o inválidos en el formulario, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	El curso se guardó en la base de datos		

Tabla 5-2: Caso de uso - Consultar curso

No. Caso	1002	**Prioridad**	Normal
Nombre	Consultar cursos		
Actor	Administrador		
Objetivo	Consultar datos de un curso registrado		
Pre-condición			
Flujo ideal	Actor: 2) El administrador selecciona o ingresa el curso a consultar.	Sistema: 1) Despliega sección de consulta de curso. 3) Muestra toda la información del curso seleccionado.	
Flujo alterno	Actor: 3.1) El administrador tiene campos vacíos y/o inválidos en el cuadro de consulta, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	Se muestra toda la información de un curso		

Tabla 5-3: Caso de uso - Editar curso

No. Caso	1003	**Prioridad**	Normal
Nombre	Editar cursos		
Actor	Administrador		
Objetivo	Modificar datos de un curso registrado		
Pre-condición	Debe existir al menos un curso registrado en el sistema		
Flujo ideal	Actor: 2) Caso de Uso 1002. 3) El administrador selecciona el curso a modificar 4) Se ingresan o modifican los datos a cambiar.	Sistema: 1) Despliega sección de modificar el curso 5) Guarda la nueva información del curso modificado	
Flujo alterno	Actor: 4.1) El administrador tiene campos vacíos y/o inválidos en el cuadro modificar, se muestra un mensaje notificando la corrección de estos.	Sistema: 1.1) No se inicia la sección editar curso, ya que, no existe al menos un curso creado.	
Pos-condición	Se modifica la información de un curso		

Tabla 5-4: Caso de uso - Eliminar curso

No. Caso	1004	**Prioridad**	Normal
Nombre	Eliminar cursos		
Actor	Administrador		
Objetivo	Eliminar datos de un curso registrado		
Pre-condición	Debe existir al menos un curso registrado en el sistema		
Flujo ideal	Actor: 2) Caso de Uso 1002. 3) El administrador selecciona el curso a eliminar.	Sistema: 1) Despliega sección eliminar curso. 4) Se elimina toda la información del curso seleccionado.	
Flujo alterno	Actor:	Sistema: 1.1) No se inicia la sección eliminar curso, ya que, no existe al menos un curso creado.	
Pos-condición	Se elimina toda la información de un curso		

Tabla 5-5: Caso de uso - Agregar programa

No. Caso	1101	**Prioridad**	Normal
Nombre	Agregar programa		
Actor	Administrador		
Objetivo	Guardar los datos de los programas		
Pre-condición	Debe existir al menos un curso registrado		
Flujo ideal	Actor: 2) El administrador rellena el formulario con los datos solicitados. 3) El administrador selecciona la opción guardar los datos.	Sistema: 1) El sistema despliega una ventana con el formulario de registro. 4) El sistema almacena en la base de datos, la información ingresada por el administrador. 5) El sistema muestra una ventana de confirmación de los datos almacenados.	
Flujo alterno	Actor: 3.1) El administrador tiene campos vacíos y/o inválidos en el formulario, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	El programa se guardó en la base de datos		

Tabla 5-6: Caso de uso - Consultar programa

No. Caso	1102	**Prioridad**	Normal
Nombre	Consultar programa		
Actor	Administrador		
Objetivo	Consultar datos de un programa registrado		
Pre-condición			
Flujo ideal	Actor: 2) El administrador selecciona o ingresa el programa a consultar.	Sistema: 1) Despliega sección de consulta de programa. 3) Muestra toda la información del programa seleccionado.	
Flujo alterno	Actor: 3.1) El administrador tiene campos vacíos y/o inválidos en el cuadro de consulta, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	Se muestra toda la información de un programa		

Tabla 5-7: Caso de uso - Editar programa

No. Caso	1103	**Prioridad**	Normal
Nombre	Editar programa		
Actor	Administrador		
Objetivo	Modificar datos de un programa registrado		
Pre-condición	Debe existir al menos un programa registrado en el sistema		
Flujo ideal	Actor: 2) Caso de Uso 1102. 3) El administrador selecciona el programa a modificar 4) Se ingresan o modifican los datos a cambiar.	Sistema: 1) Despliega sección de modificar el programa 5) Guarda la nueva información del programa modificado	
Flujo alterno	Actor: 4.1) El administrador tiene campos vacíos y/o inválidos en el cuadro modificar, se muestra un mensaje notificando la corrección de estos.	Sistema: 1.1) No se inicia la sección editar programa, ya que, no existe al menos un programa creado.	
Pos-condición	Se modifica la información de un programa		

Tabla 5-8: Caso de uso - Eliminar programa

No. Caso	1104	**Prioridad**	Normal
Nombre	Eliminar programa		
Actor	Administrador		
Objetivo	Eliminar datos de un programa registrado		
Pre-condición	Debe existir al menos un programa registrado en el sistema		
Flujo ideal	Actor: 2) Caso de Uso 1102. 3) El administrador selecciona el programa a eliminar.	Sistema: 1) Despliega sección eliminar programa. 4) Se elimina toda la información del programa seleccionado.	
Flujo alterno	Actor:	Sistema: 1.1) No se inicia la sección eliminar programa, ya que, no existe al menos un programa creado.	
Pos-condición	Se elimina toda la información de un programa		

Tabla 5-9: Caso de uso - Agregar usuario

No. Caso	1201	**Prioridad**	Normal
Nombre	Agregar usuario		
Actor	Administrador		
Objetivo	Guardar los datos de los usuarios		
Pre-condición			
Flujo ideal	Actor: 2) El administrador rellena el formulario con los datos solicitados. 3) El administrador selecciona la opción guardar los datos.	Sistema: 1) El sistema despliega una ventana con el formulario de registro. 4) El sistema almacena en la base de datos, la información ingresada por el administrador. 5) El sistema muestra una ventana de confirmación de los datos almacenados.	
Flujo alterno	Actor: 3.1) El administrador tiene campos vacíos y/o inválidos en el formulario, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	El usuario se guardó en la base de datos		

Tabla 5-10: Caso de uso - Consultar usuario

No. Caso	1202	**Prioridad**	Normal
Nombre	Consultar usuario		
Actor	Administrador		
Objetivo	Consultar datos de un usuario registrado		
Pre-condición			
Flujo ideal	Actor: 2) El administrador selecciona o ingresa el usuario a consultar.	Sistema: 1) Despliega sección de consulta de usuario. 3) Muestra toda la información del usuario seleccionado.	
Flujo alterno	Actor: 3.1) El administrador tiene campos vacíos y/o inválidos en el cuadro de consulta, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	Se muestra toda la información de un usuario		

Tabla 5-11: Caso de uso - Editar usuario

No. Caso	1203	**Prioridad**	Normal
Nombre	Editar usuario		
Actor	Administrador		
Objetivo	Modificar datos de un usuario registrado		
Pre-condición	Debe existir al menos un usuario registrado en el sistema		
Flujo ideal	Actor: 2) Caso de Uso 1202. 3) El administrador selecciona el usuario a modificar 4) Se ingresan o modifican los datos a cambiar.	Sistema: 1) Despliega sección de modificar el usuario. 5) Guarda la nueva información del usuario modificado.	
Flujo alterno	Actor: 4.1) El administrador tiene campos vacíos y/o inválidos en el cuadro modificar, se muestra un mensaje notificando la corrección de estos.	Sistema: 1.1) No se inicia la sección editar usuario, ya que, no existe al menos un usuario creado.	
Pos-condición	Se modifica la información de un usuario		

Tabla 5-12: Caso de uso - Eliminar Usuario

No. Caso	1204	**Prioridad**	Normal
Nombre	Eliminar usuario		
Actor	Administrador		
Objetivo	Eliminar datos de un usuario registrado		
Pre-condición	Debe existir al menos un usuario registrado en el sistema		
Flujo ideal	Actor: 2) Caso de Uso 1202. 3) El administrador selecciona el usuario a eliminar.	Sistema: 1) Despliega sección eliminar usuario. 4) Se elimina toda la información del usuario seleccionado.	
Flujo alterno	Actor:	Sistema: 1.1) No se inicia la sección eliminar usuario, ya que, no existe al menos un usuario creado.	
Pos-condición	Se elimina toda la información de un usuario		

Tabla 5-13: Caso de uso - Agregar requisito

No. Caso	1301	**Prioridad**	Normal
Nombre	Agregar requisito		
Actor	Gestor		
Objetivo	Guardar los datos de los requisitos		
Pre-condición	Debe existir al menos un programa registrado		
Flujo ideal	Actor: 2) El gestor rellena el formulario con los datos solicitados. 3) El gestor selecciona la opción guardar los datos.	Sistema: 1) El sistema despliega una ventana con el formulario de registro. 4) El sistema almacena en la base de datos, la información ingresada por el administrador. 5) El sistema muestra una ventana de confirmación de los datos almacenados.	
Flujo alterno	Actor: 3.1) El gestor tiene campos vacíos y/o inválidos en el formulario, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	El requisito se guardó en la base de datos		

Tabla 5-14: Caso de uso - Consultar requisito

No. Caso	1302	**Prioridad**	Normal
Nombre	Consultar requisito		
Actor	Gestor		
Objetivo	Consultar datos de un requisito registrado		
Pre-condición			
Flujo ideal	Actor: 2) El gestor selecciona o ingresa el requisito a consultar.	Sistema: 1) Despliega sección de consulta de requisito. 3) Muestra toda la información del requisito seleccionado.	
Flujo alterno	Actor: 3.1) El gestor tiene campos vacíos y/o inválidos en el cuadro de consulta, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	Se muestra toda la información de un requisito		

Tabla 5-15: Caso de uso - Editar requisito

No. Caso	1303	**Prioridad**	Normal
Nombre	Editar requisito		
Actor	Gestor		
Objetivo	Modificar datos de un requisito registrado		
Pre-condición	Debe existir al menos un requisito registrado en el sistema		
Flujo ideal	Actor: 2) Caso de Uso 1302. 3) El gestor selecciona el requisito a modificar 4) Se ingresan o modifican los datos a cambiar.	Sistema: 1) Despliega sección de modificar el requisito. 5) Guarda la nueva información del requisito modificado.	
Flujo alterno	Actor: 4.1) El gestor tiene campos vacíos y/o inválidos en el cuadro modificar, se muestra un mensaje notificando la corrección de estos.	Sistema: 1.1) No se inicia la sección editar requisito, ya que, no existe al menos un requisito creado.	
Pos-condición	Se modifica la información de un requisito		

Tabla 5-16: Caso de uso - Analizar documento

No. Caso	1401	**Prioridad**	Normal
Nombre	Analizar documento		
Actor	Gestor		
Objetivo	Ejecuta el análisis del documento y guarda los resultados		
Pre-condición	Debe existir al menos un requisito registrado		
Flujo ideal	Actor: 2) El gestor carga el documento y selecciona los requisitos. 3) El gestor inicia el análisis del documento.	Sistema: 1) El sistema despliega una ventana con el formulario de análisis. 4) El sistema analiza el documento y guarda los resultados en la base de datos. 5) El sistema muestra una ventana de confirmación de los datos almacenados.	
Flujo alterno	Actor: 3.1) El gestor tiene campos vacíos y/o inválidos en el formulario, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	El documento se analizó y se guardó el resultado en la base de datos		

Tabla 5-17: Caso de uso - Consultar resultados

No. Caso	1501	**Prioridad**	Normal
Nombre	Consultar resultado		
Actor	Gestor / Consulta		
Objetivo	Consultar datos de un resultado almacenado		
Pre-condición			
Flujo ideal	Actor: 2) El gestor/consulta selecciona o ingresa los datos del resultado a consultar.	Sistema: 1) Despliega sección de consulta de resultados. 3) Muestra toda la información del resultado seleccionado.	
Flujo alterno	Actor: 3.1) El gestor/consulta tiene campos vacíos y/o inválidos en el cuadro de consulta, se muestra un mensaje notificando la corrección de estos.	Sistema:	
Pos-condición	Se muestra toda la información de un resultado de análisis		

Tabla 5-18: Caso de uso - Visualizar datos

No. Caso	1601	**Prioridad**	Normal
Nombre	Visualizar datos		
Actor	Administrador		
Objetivo	Generar un informe gráfico de los resultados de análisis		
Pre-condición	Debe existir al menos un resultado registrado en el sistema		
Flujo ideal	Actor: 2) Caso de Uso 1401/1501. 3) El gestor/consulta selecciona la opción visualizar.	Sistema: 1) Despliega la opción visualizar resultados. 4) Se despliega el informe gráfico de resultados.	
Flujo alterno	Actor:	Sistema:	
Pos-condición			

5.2. Diagrama Entidad-Relación

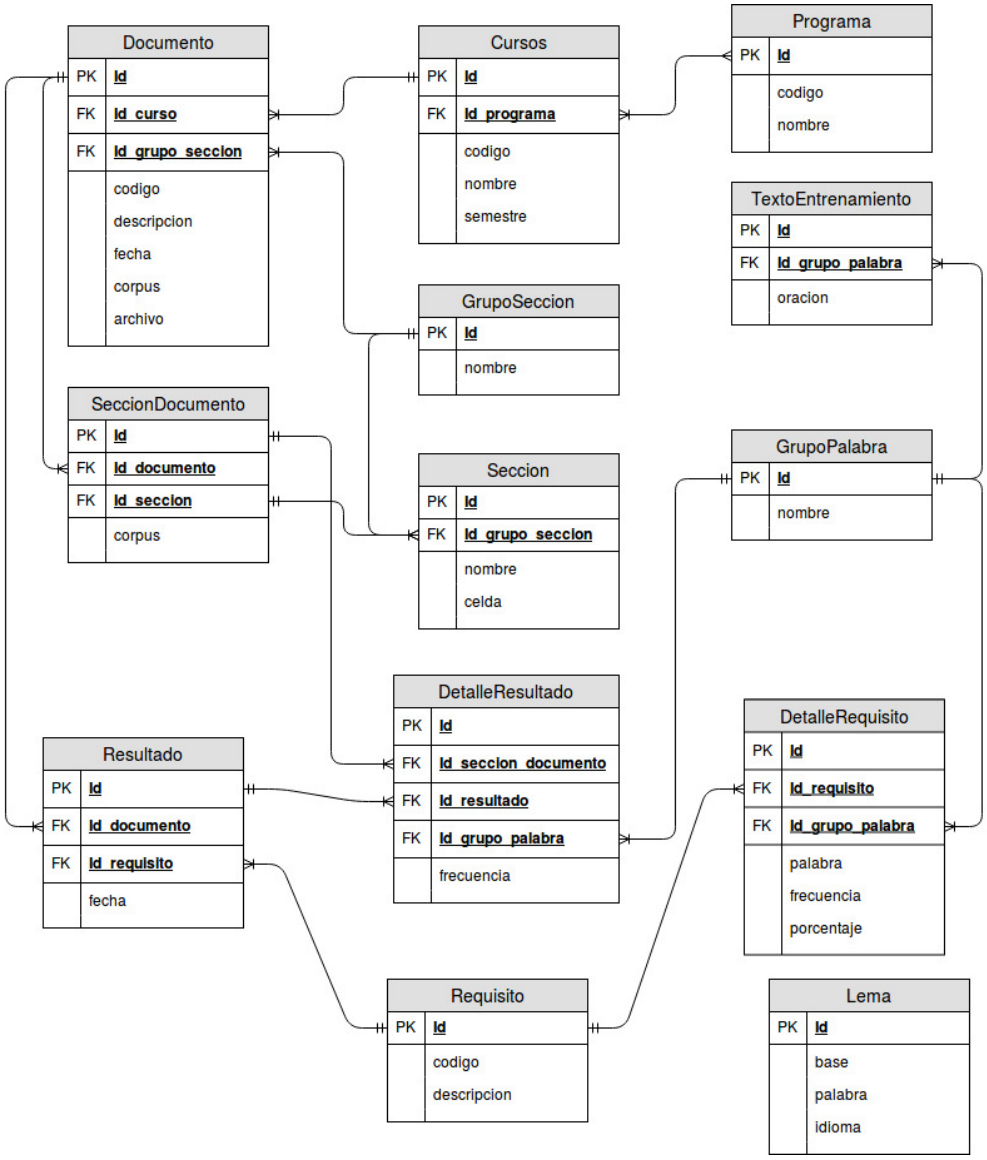

Figura 5-2: Diagrama Entidad-Relación

El diagrama Entidad-Relación es el modelo conceptual más usado para el diseño conceptual de bases de datos. Se encuentra formado por un conjunto de conceptos que permite la descripción de la realidad mediante un conjunto de representaciones gráficas y lingüísticas. Este tipo de diagramas se realizan para llevar a cabo el diseño y la posterior depuración de la base de datos relacional. A continuación, en la figura **5-2** podemos presenciar la elaboración de nuestro diagrama entidad-relación, en el cual se puede evidenciar como las entidades que tenemos en nuestro sistema tienen sus relaciones entre sí. En este diagrama se puede evidenciar que existe un programa,

el cual contiene determinados cursos que a su vez posee un documento, el cual se encuentra dividido en secciones para que sea mucho más fácil su interpretación. Cada sección posee un grupo de palabras, mediante las cuales podemos realizar un entrenamiento al sistema para que este posteriormente permita obtener unos resultados de forma detallada, teniendo en cuenta que todo esto se hace basándose en unos requisitos previamente descritos tomando como ayuda la taxonomía de Bloom. Habiendo obtenido cada uno de los resultados por sección, el documento recoge cada uno de estos para al final poder mostrar al usuario un resultado general.

5.3. Diagrama de actividades

Los diagramas de actividades complementan los casos de uso al proporcionar una representación gráfica del flujo de interacciones dentro de un escenario específico. Este tipo de diagrama, es básicamente un diagrama de flujo, ya que en su estructura muestra como fluye el control de unas clases a otras con el fin de culminar con un flujo de control total que encuentra correspondencia con la obtención de un proceso más complejo. En el diagrama que usa o en que se basó nuestro sistema, conlleva a la carga de un documento, del cual se hará extracción del corpus para así poder pre-procesar dicha información que contenga. Luego de haber realizado este proceso, se genera una bolsa de palabras que permitirá por medio de estas establecer y a su vez cargar los requisitos por parte del usuario, posterior a esto debe analizar las frecuencias que contenga las bolsas de palabras que previamente fueron creadas para así poder comparar y generar los resultados.

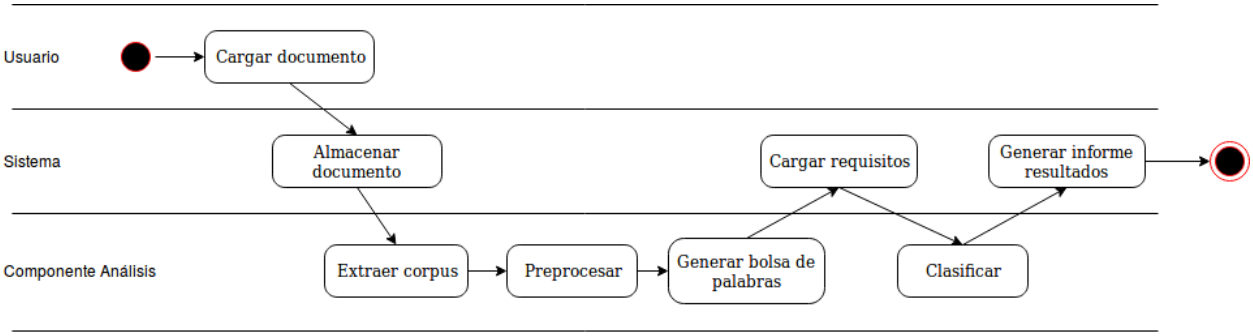

Figura 5-3: Diagrama de actividades - Análisis documentos

5.4. Diagrama de componentes

Los elementos de este tipo de diagramas son los componentes del software y cada uno de los tipos de dependencia que existe entre cada uno de ellos. El diagrama de componentes ilustra las piezas que contienen el software, los controladores embebidos, etc. que conformaran el sistema, esto quiere decir, que es el que nos suministra la visión física de la construcción del sistema de información. En la figura **5-4** podemos encontrar el diagrama de nuestro sistema, el cual contiene

4 componentes establecidos entre los que se encuentran un servidor web, una aplicación web que a su vez realiza la conexión a la base de datos del sistema, y por ultimo un componente de análisis de documentos. Es importante destacar que el componente de análisis de documentos fue creado de manera independiente para que pueda ser importado en distintos proyectos de la misma especie.

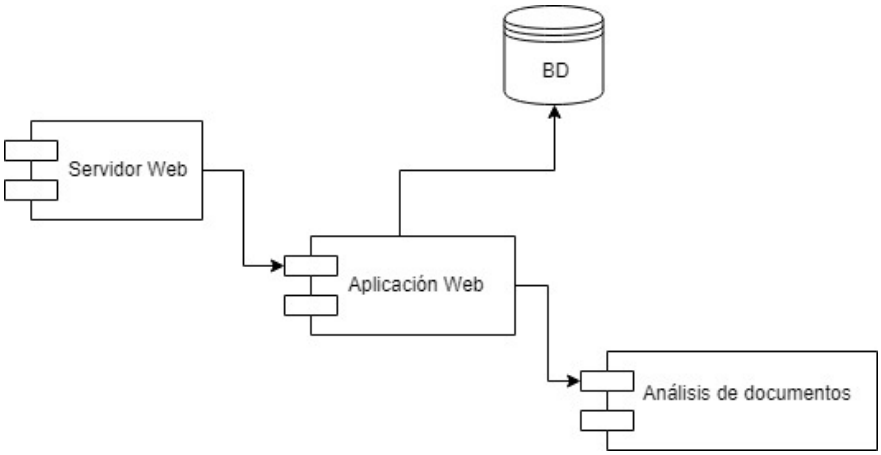

Figura 5-4: Diagrama de componentes

6 Desarrollo

A continuación se describen los componentes del producto a desarrollar. Específicamente el aplicativo Web y el componente de análisis y procesamiento de texto. Se detalla entonces el funcionamiento de cada uno, se describe la interacción que existe entre ellos y se especifica como han sido implementados. Junto a esto se listan los lenguajes, librerías y herramientas que apoyan la construcción de cada uno de ellos. Iniciamos con el componente de análisis que hace parte del núcleo de la solución y posteriormente con el aplicativo Web que gestiona la interacción con el usuario final.

6.1. Componente de Análisis de documentos

Es el encargado de realizar la minería del texto. Se apoya en el procesado de lenguaje natural para manipular el texto. Aplica las técnicas pertinentes para identificar las palabras, procesar el texto y extraer la información relevante. Es decir, que toma el corpus en crudo, realiza un trabajo de pre-procesamiento y se obtiene una forma intermedia llamada bolsa de palabras. La cual tiene la forma más óptima para el descubrimiento de conocimiento y la clasificación. Esta última se realiza con base en la Taxonomía de Bloom, sin embargo, dado el diseño configurable de la solución, se puede usar una taxonomía particular.
El componente cuenta diversas funciones para realizar la labor antes mencionada. Contiene métodos para eliminar signos de puntuación, para eliminar stop words, lematización de verbos, tokenización de palabras y el cálculo de frecuencia absoluta de cada palabra. La eliminación de signos se realiza mediante la aplicación de expresiones regulares. La eliminación de stop words y la tokenización de palabras se realizan con el apoyo de la librería NLTK. La lematización se realiza usando puramente código python, pero se apoya en una base de datos de lemas en español que ha sido previamente cargada.
El componente aplica un pre-procesamiento de un texto para generar una bolsa de palabras. Primero se eliminan las stop word, luego se eliminan los signos de puntuación, en tercer lugar se realiza la tokenización y finalmente se aplica la lematización. Cabe resaltar que la configuración del pre-procesado es particular a este proyecto. El resultado obtenido se puede expresar de la siguiente manera:

$$BDP = [w_i \dots w_n]$$

Donde *BDP* es el conjunto bolsa de palabras, *w* representa a las palabras e *i* es el índice de cada

palabra. Para generar una forma final mas útil para el análisis, se calcula la frecuencia de cada palabra. De esta manera se obtiene un nuevo conjunto de la siguiente forma:

$$BDP = [(w_i, F(w_i)) \dots (w_n, F(w_n))]$$

Donde F(w) representa la frecuencia de cada palabra dentro de la bolsa. Este nuevo conjunto permite al componente realizar un análisis contra un conjunto de requisitos. Los cuales, para este caso, representan el conjunto de verbos de la Taxonomía de Bloom, clasificados según el nivel al que pertenecen. Dicho análisis consiste en detectar la presencia de los verbos dentro de un texto y su incidencia conforme al volumen del mismo.
Para detectar la existencia de los verbos e identificar a que nivel pertenecen, se usa un clasificador Naive Bayes. El cual toma los datos de entrenamiento, previamente cargados, para crear un modelo que se usará para la clasificación. Para entrenar el clasificador se usa un conjunto de oraciones, ya clasificados en la taxonomía de Bloom, el cual contiene varias combinaciones de los verbos para cada nivel. Con el clasificador listo se ingresan las bolsas de palabras y se obtiene un nivel de la taxonomía. Las bolsas representan los datos extraídos de cada sección del syllabus. Estas a su vez se dividen en oraciones. Luego se toma cada una de ellas y se clasifica. Al finalizar se resume que niveles hacen presencia en cada sección. De manera general el proceso de análisis y clasificación se resume en la siguiente figura **6-1**.

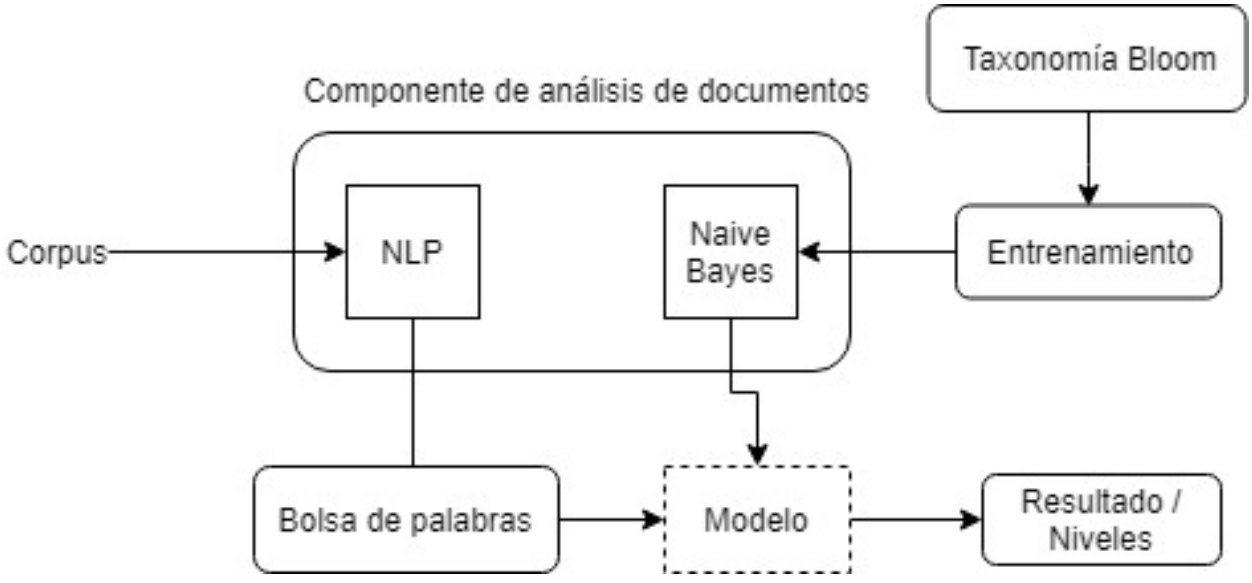

Figura 6-1: Proceso de análisis y clasificación

Este componente fue desarrollado enteramente en python. El cual es un lenguaje muy productivo, flexible, fácil de depurar y testear [20]. Además, actualmente, en una de las mejores opciones para implementar sistemas de inteligencia artificial que, dependiendo de su tipo, requieren un conjunto de librerías de apoyo. Para este caso se usó NLTK por su reputación y sencilla implementación. Esta librería se enfoca en procesamiento de lenguaje natural, pero se extiende y abarca funciones

de clasificación de texto, entre otras. Con ella se implementan los métodos de pre-prosesamiento del corpus y el clasificador Nayve Bayes. Este último se basa en el teorema de Bayes, es muy eficiente y se caracteriza por su fácil entrenamiento, además es multipropósito [21].

6.2. Aplicación Web

Se encarga de gestionar la persistencia de los datos y de la ejecución del análisis de documentos. Está compuesta de dos partes. Una de administración y configuración y un front-end para la carga y procesamiento de documentos. Se ha construido con base en los requisitos acordados y conforme al diseño presentado previamente. La vista administrativa ofrece una operación sencilla y común en todos los aspectos a configurar. Es decir, para todos ellos las operaciones de registro, consulta, edición y eliminación son iguales. Por otra parte, la vista de análisis ofrece opciones paso a paso y de consulta que no presentan dificultad para su aprendizaje.

6.2.1. Administración y Configuración

La vista de administración cuenta con funciones de registro, consulta y edición de los aspectos de configuración. Es decir, las operaciones CRUD de las tablas requeridas para el funcionamiento del aplicativo. Dentro de los aspectos a configurar se incluyen la gestión de programas y sus respectivos cursos. Estos deben estar cargados, debido a que se asociarán con los documentos a analizar. Dicho análisis se gestiona en base a tres aspectos mas, los requisitos, los detalles de esos requisitos y los grupos de palabras asociados a los detalles. En síntesis, hablando de taxonomía de Bloom, estos últimos representan los niveles y su respectivo conjunto de verbos. Finalmente cabe resaltar que los datos de configuración pueden modificarse o extenderse, según lo que se desea analizar y la forma de hacerlo.

Las operaciones de esta vista se han construido con el framework Django [22]. Este cuenta con modulo de administración, el cual está inmerso en la construcción de los proyectos. Resulta muy útil para operaciones básicas de configuración. Con establecer algunos parámetros a un conjunto de tablas, podemos obtener una vista con operaciones CRUD. Estas ya se encuentra previamente testeadas, son confiables y con una funcionalidad sencilla. Manejan y controlan las relaciones entre tablas y permite personalizar los filtros de búsqueda. Finalmente realiza el control de acceso de usuarios administrativos. Dadas todas estas características fue la opción escogida para la vista de administración.

A continuación se describen las operaciones principales de esta vista. Para hacer uso de ella se requiere de autenticación, es decir que el acceso está restringido mediante usuarios de tipo administrativo. El acceso a esta vista se encuentra también en la vista de operaciones, pero solo los usuarios administrativos pueden acceder a ella. El siguiente grupo de imágenes muestran la operación de la vista administrativa. Se puede notar el funcionamiento del CRUD en las tablas. Cabe recordar que estas son solo las de configuración, las demás son manipuladas desde la vista de operación.

La figura **6-2** muestra el menú principal de administración. En esta se relacionan todas las tablas de configuración. Al hacer clic en cada una de ellas se accede a las operaciones CRUD de la misma. Además de los datos a configurar, se pueden crear los usuarios y los grupos que pueden acceder a la aplicación.

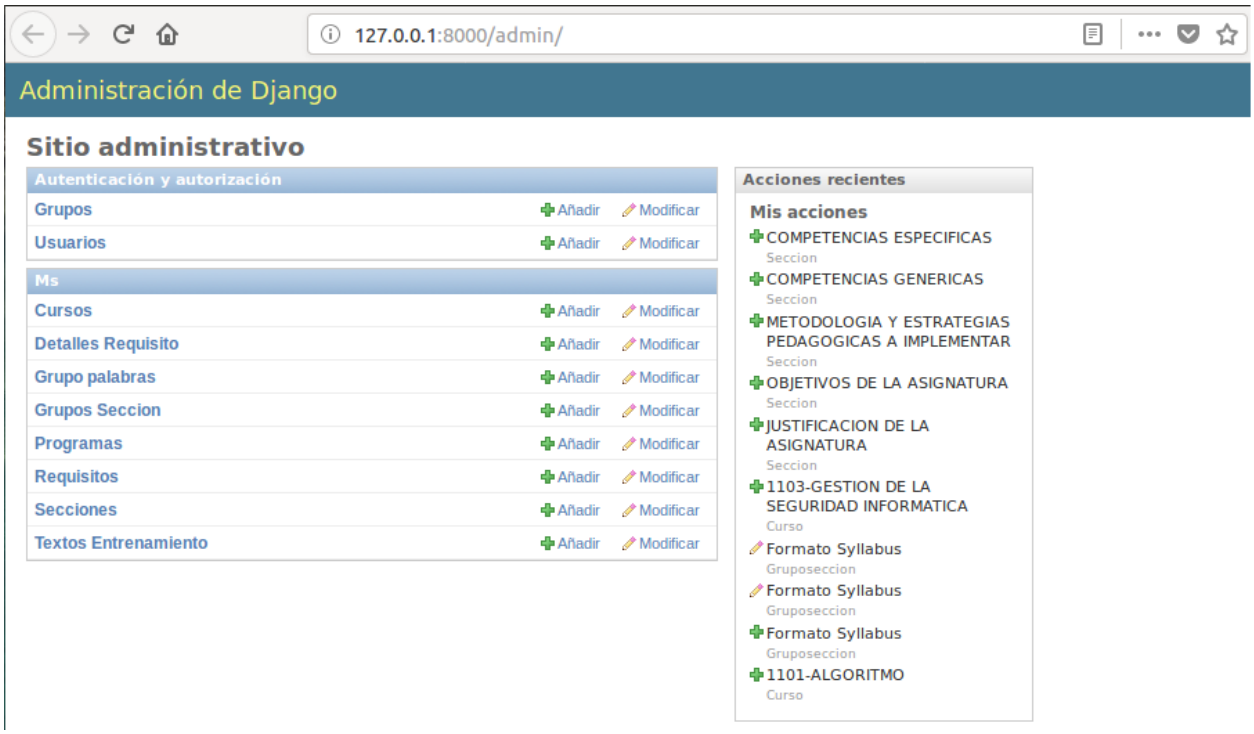

Figura 6-2: Vista de administración

Si desde el menú de administración hacemos clic en *Programas*, entonces se puede ver el listado de los programas registrados en la base de datos (figura **6-3**). Si se desea crear uno nuevo se debe hacer clic en *agregar* en la parte superior. La figura **6-4** muestra el formulario para almacenar los programas. Si se desea editar un programa existente, entonces se debe hacer clic a uno de ellos dentro de la tabla de consulta. La figura **6-5** muestra el formulario de edición de programas. Este último tiene, en la parte inferior, la opción de eliminar el registro. Dicha eliminación dependerá de las relaciones con otras tablas.

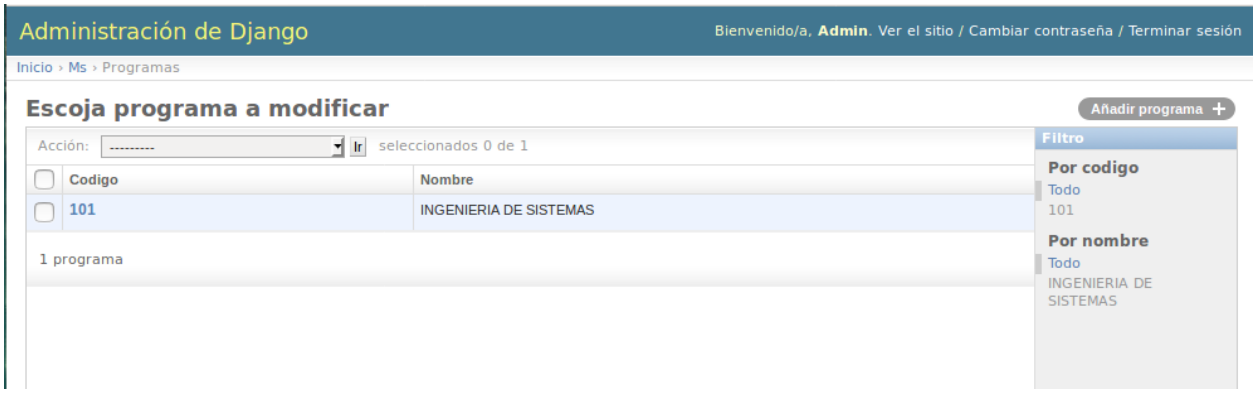

Figura 6-3: Listar programas

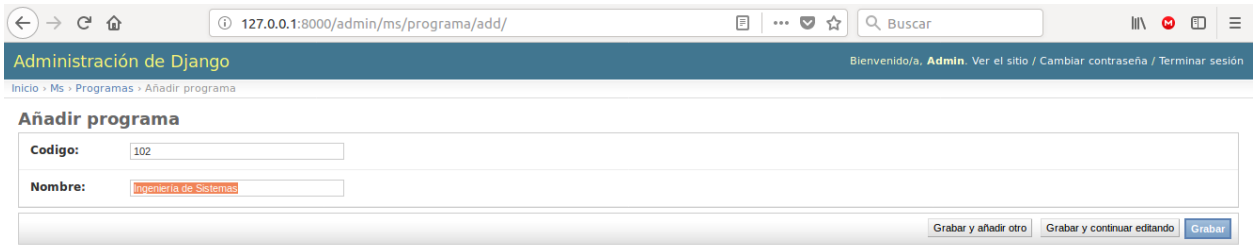

Figura 6-4: Crear programas

Figura 6-5: Editar programas

Los programas se relacionan con un conjunto de cursos. Cabe resaltar que los cursos pueden pertenecer a varios programas. Por ejemplo las matemáticas pueden pertenecer a varias ingenierías. El listado de cursos se puede ver en la figura **6-6**. Desde la consulta se pueden agregar nuevos cursos haciendo clic en la primera columna. Esto nos lleva al formulario de edición de cursos como se ve en la figura **6-7**, aunque también se pueden agregar nuevos, usando el botón *agregar curso*. Al editar un curso también, de ser necesario, se puede agregar un nuevo programa, ver figura **6-8**.

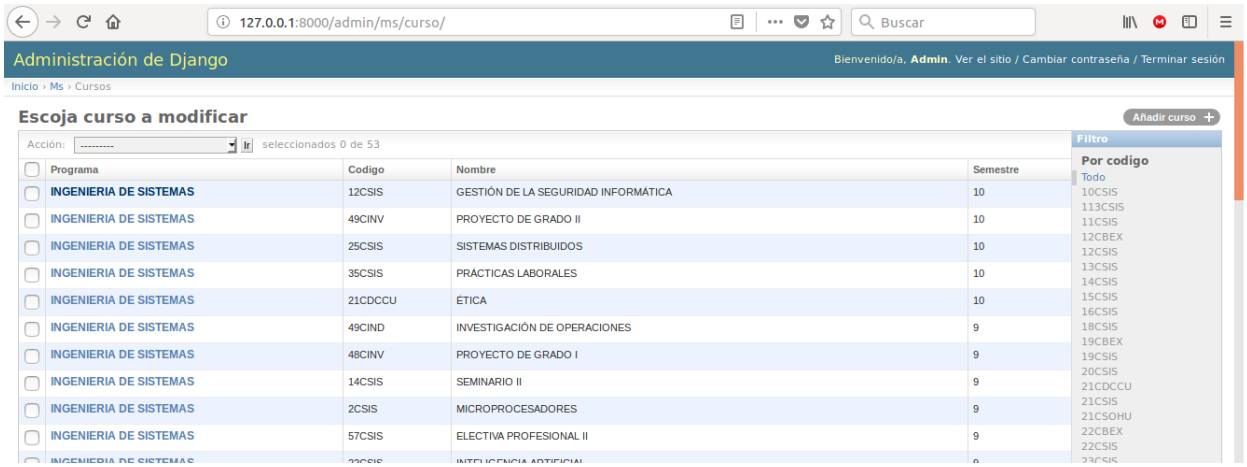

Figura 6-6: Listar cursos

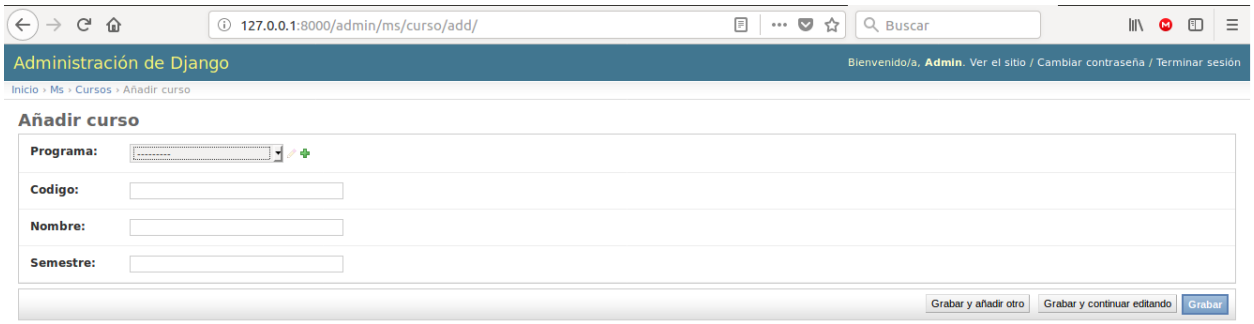

Figura 6-7: Crear cursos

Figura 6-8: Agregar programa desde curso

Al igual que los cursos y programas, la secciones, que definen la estructura del syllabus, funciona igual que los programas. En la figura **6-9** se ve el listado de secciones. De la misma forma que

el anterior usamos la primera columna para editar, ver figura **6-10**. Así como los cursos y los programas, las secciones se relacionan con un *grupo sección*, que para este caso representa la justificación, objetivos, competencias y demás secciones del syllabus.

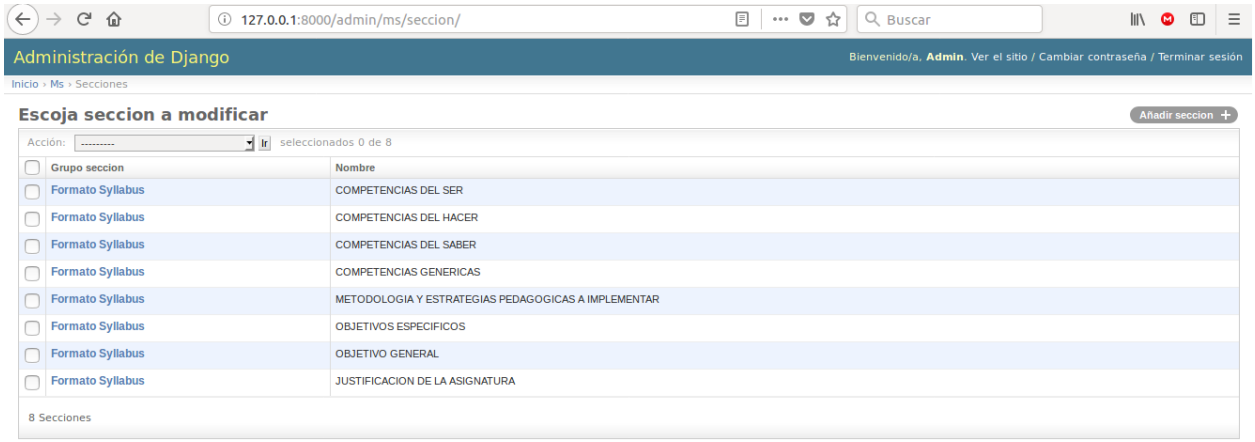

Figura 6-9: Listar secciones

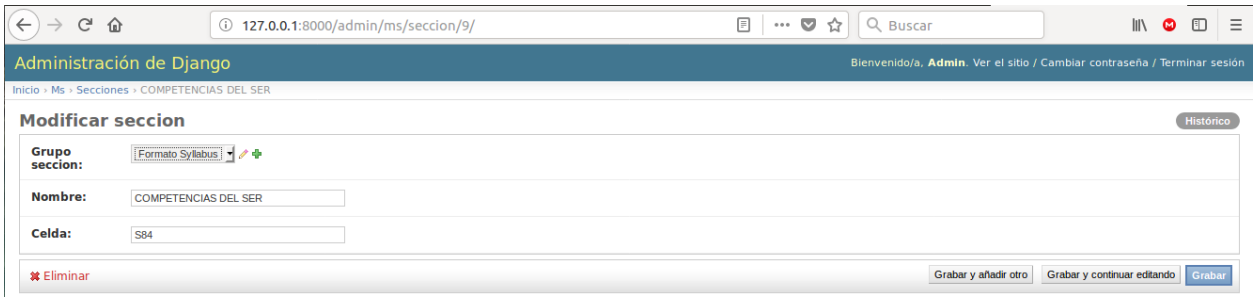

Figura 6-10: Editar secciones

Los grupos de palabras representan los niveles de la taxonomía de Bloom. Están presentes en los textos de entrenamiento y acompañan a los verbos almacenados en los detalles de requisitos. En la figura **6-11** se muestra la tabla de consulta de grupos de palabras. No solo se pueden ver sus datos, sino también editarlos haciendo clic en cualquier palabra de la lista. También se pueden crear nuevos como se ve en la figura **6-12**.

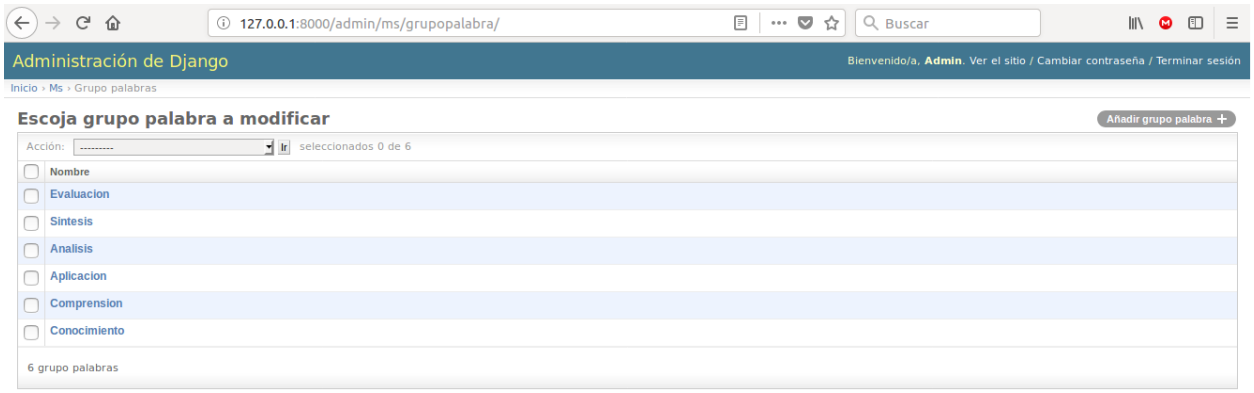

Figura 6-11: Listar grupo palabras

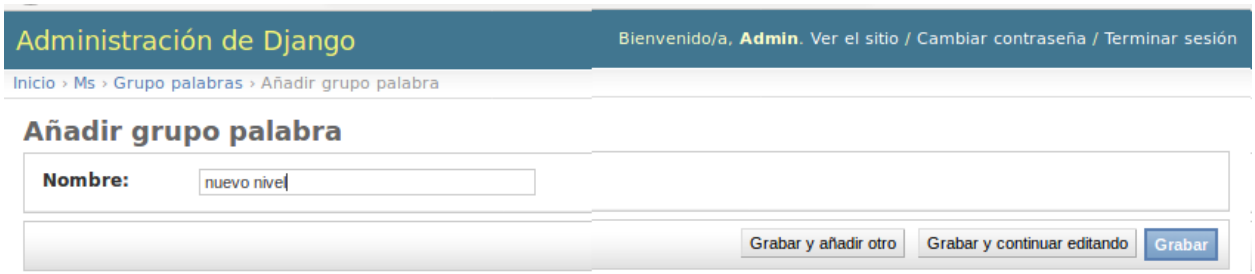

Figura 6-12: Crear grupo palabras

Los requisitos representan los criterios para el análisis del texto. En la figura **6-13** muestra la consulta de requisitos, en ella se puede ver el registro de *Taxonomía de Bloom*, el cual es usado en el presente proyecto. El conjunto de verbos relacionados se guarda en *detalles requisito*. La figura **6-14** muestra la consulta de detalles requisito. En ella se puede ver el grupo de palabra relacionado. Como ya se ha mencionado los grupos de palabras son los niveles. Cabe resaltar que estos ya deben existir en la base de datos, de otra manera no se pueden registrar los verbos.

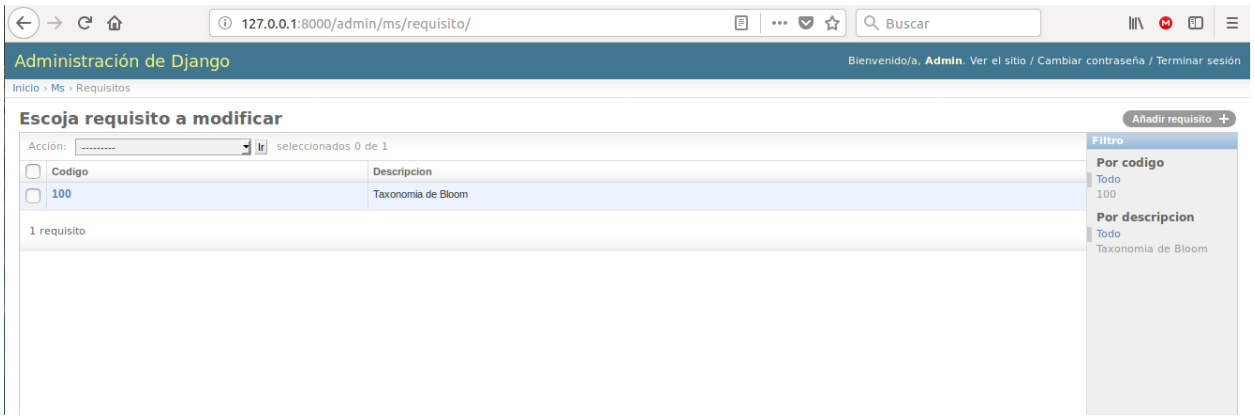

Figura 6-13: Listar requisitos

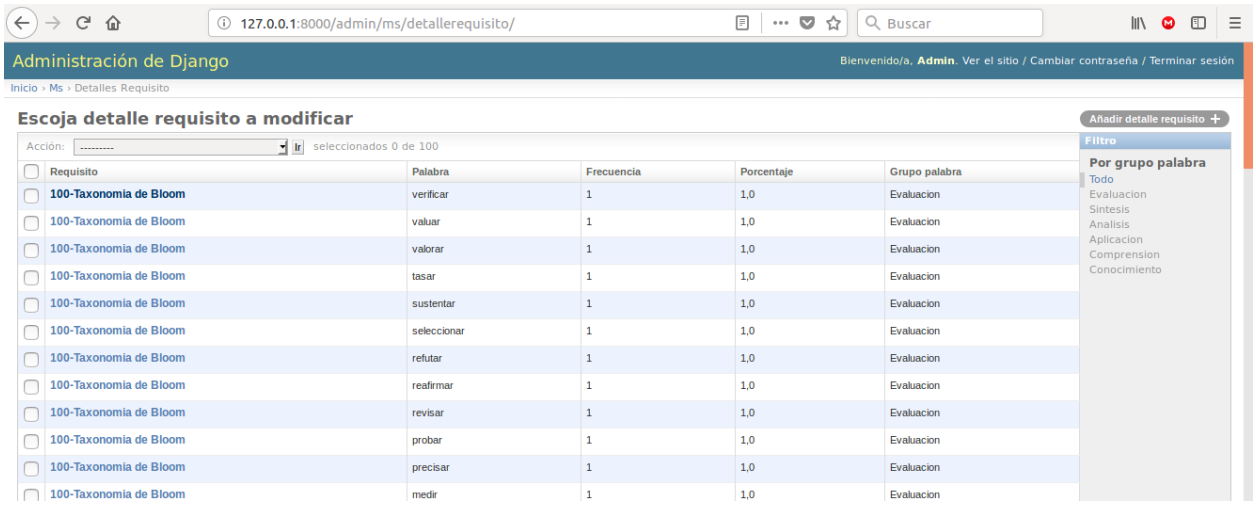

Figura 6-14: Listar detalles requisitos

Finalmente se debe resaltar que hay que tener especial cuidado con la configuración. Pues esta determina el buen comportamiento del aplicativo, pero también puede extender sus capacidades. Es decir se pueden realizar otros tipos de análisis o cargar documentos con diferentes estructuras.

6.2.2. Operación

La vista de operación cuenta con un inicio de sesión para autenticar usuarios. Solo se requiere un usuario de nivel general, aunque es posible autenticarse con un usuario de administración. Una vez se ingresa se presentan tres opciones, nuevo, consulta y administración. Esta última nos lleva a la vista de opciones descrita en la sección anterior. Solo se accede si el usuario es de tipo administrador. Este debe ser configurado desde la vista de administración y configuración. Dentro de las opciones nuevo y consulta, se incluye el proceso de análisis y generación de reportes los cuales se describirán mas adelante.

La figura **6-15** muestra la pantalla de inicio de sesión. Se debe suministrar un usuario y una contraseña para ingresar. Si se hace clic en el botón de enviar, entonces se realiza la autenticación del usuario. Si esta es exitosa el usuario ingresa al menú principal del aplicativo. Una vez dentro, se puede ver el menú del aplicativo. en la figura **6-16** se pueden ver las opciones, nuevo, consulta y administrar ya mencionadas.

Figura 6-15: Iniciar sesión

Figura 6-16: Menú del aplicativo

Para cargar nuevos documentos a analizar se da clic en la opción *Nuevo*. Se presenta un formulario como se muestra en la figura **6-17**. Lo primero es seleccionar el curso asociado al documento, luego se escoge el formato de secciones y finalmente se selecciona el documento en Excel. Al hacer clic en *Enviar* se almacena el documento en base de datos y se preprocesa.

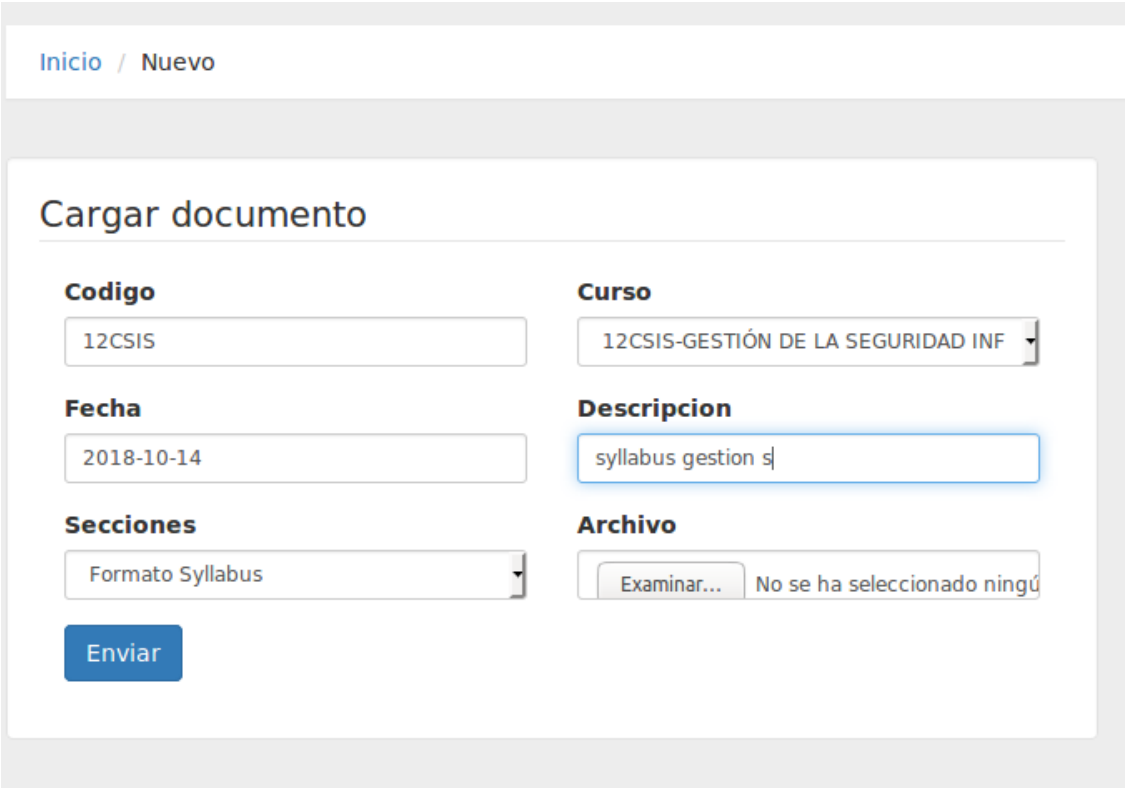

Figura 6-17: Nuevo documento

Luego del preprocesar el texto el siguiente paso es el análisis. En la figura **6-18** se ve el formulario en donde se escoge el requisito de análisis. Una vez seleccionado se da clic en *Enviar* y se inicia el proceso de minería y clasificación. Al finalizar se genera el reporte de resultados (ver figura **6-19** y **6-20**). El reporte muestra los resultados por cada sección del documento, es decir, justificación, objetivo general, objetivos específicos, etc. El primer gráfico, a la izquierda de cada sección, muestra la proporción que existe entre los niveles identificados, teniendo en cuenta la proporción que no tiene clasificación. La medida para calcular esta proporción, es el número de oraciones dentro de cada sección. La clasificación se realiza por cada oración. En el segundo gráfico, a la derecha de cada sección, se muestra cuales, de los seis niveles, se hace presente en cada sección y su respectiva proporción entre ellos.

Figura 6-18: Opciones de análisis de documento

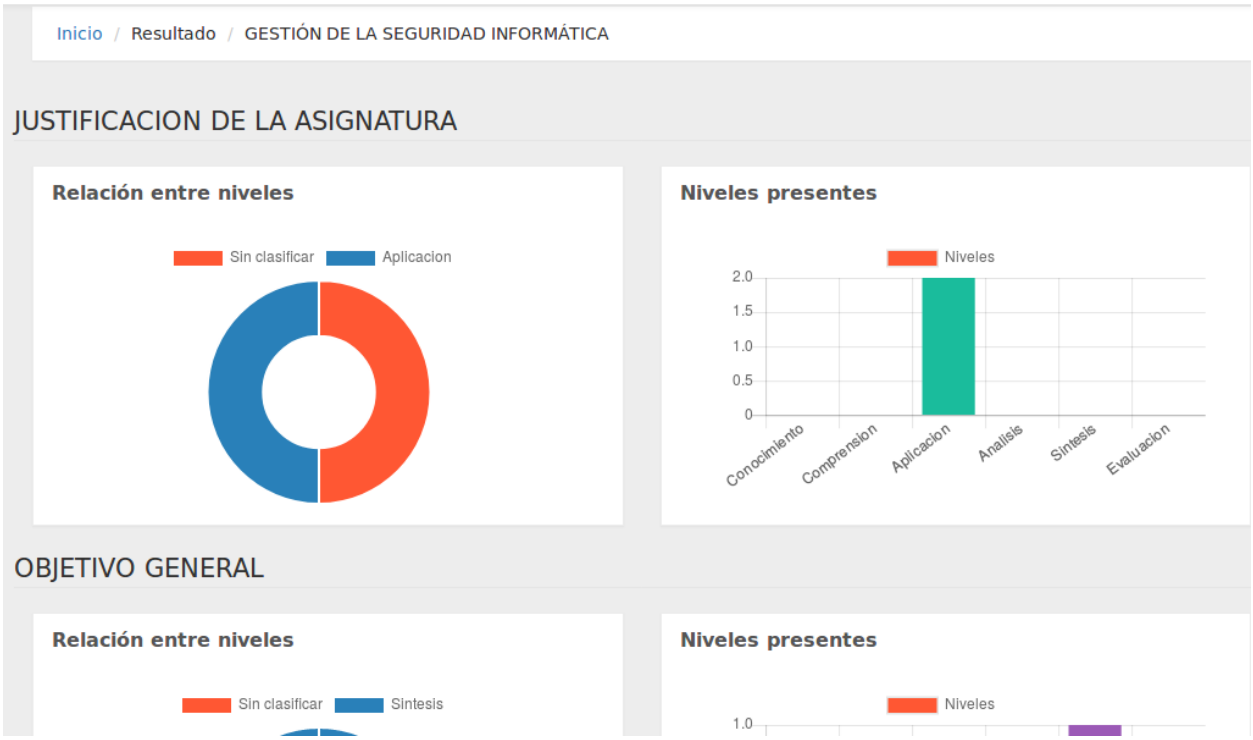

Figura 6-19: Reporte parte 1

Figura 6-20: Reporte parte 2

Con el documento cargado y los resultados del análisis ya guardados, se puede consultar y generar nuevamente el informe. Para esto hacemos clic en *Consulta* desde el menú principal. En la figura **6-21** se puede ver el formulario de consulta. se debe escoger un curso y hacer clic en *Enviar*. Con esto se genera una búsqueda en la base de datos y se genera un resultado.

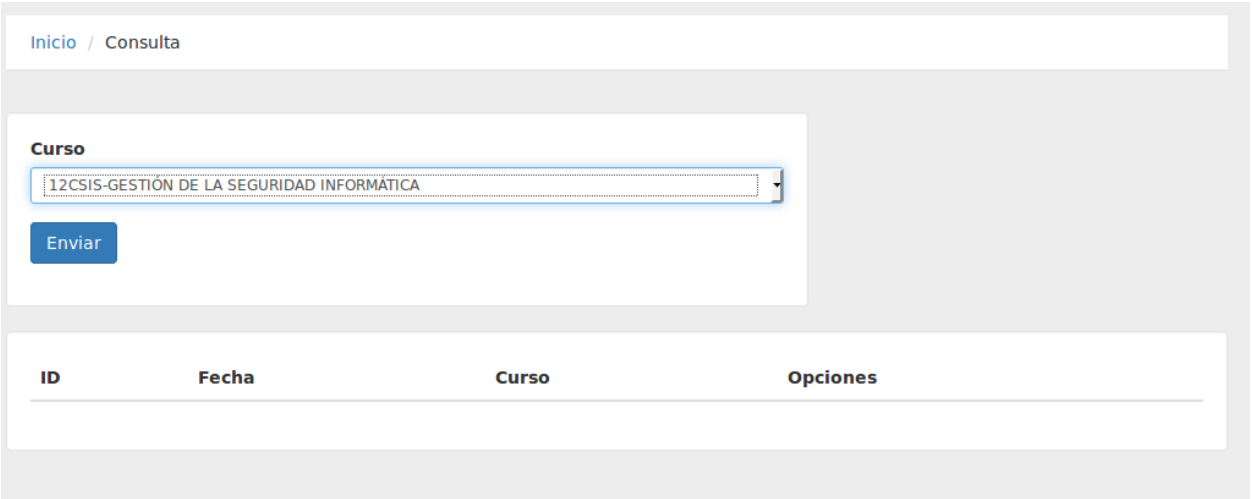

Figura 6-21: Consulta de documentos

La respuesta de una consulta típica se puede ver en la figura **6-22**. Desde ella se puede observar

una tabla con varias columnas de información. Entre ellas existe una de opción. Dar clic en esta se despliega una vista para seleccionar el análisis que se desea (ver figura **6-23**).

Figura 6-22: Consulta de documentos encontrados

Si se desea generar nuevamente el informe de un documento, se debe hacer clic en la descripción de los resultados como se ve en la figura **6-23**. Esto nos lleva a la vista de informes como se ve en las figuras **6-19** y **6-20**.

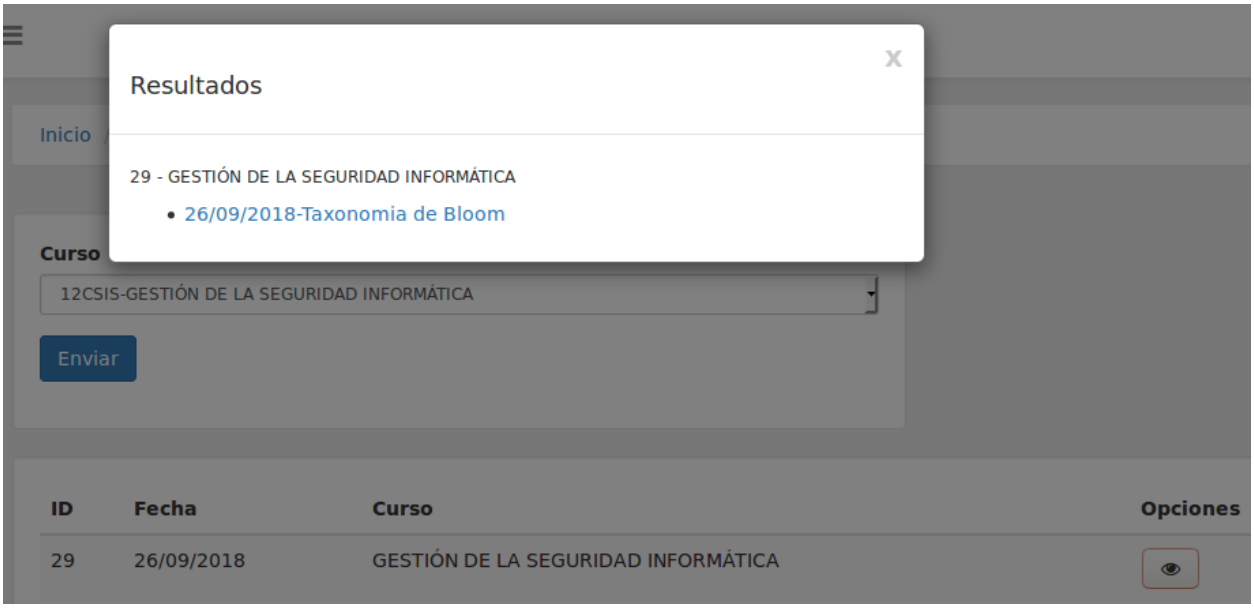

Figura 6-23: Ver resultados de análisis

7 Pruebas y Resultados

La planeación de pruebas del proyecto incluye, las pruebas modulares, las de integración y las de sistema. En las modulares se testean las clases y componentes individuales. Las de integración validan la correcta operación entre distintas partes del software. Las últimas permiten comprobar el funcionamiento general del sistema. El desarrollo ha pasado por múltiples pruebas, incluyendo las categorías ya mencionadas. A continuación se detallan las de caja negra. Se han definido los casos según el objetivo a lograr y se especifican los resultados. Para ello se ha definido una estructura que incluye, la descripción del caso, la vista, módulo o método a evaluar, las entradas, las salidas esperadas y las obtenidas. Se inicia con el componente de análisis de texto y finalizamos con las pruebas generales del sistema.

7.1. Pruebas modulares

El siguiente grupo de pruebas abarca el componente de análisis de texto. Este cuenta con una clase analizador y una clasificador. Las pruebas se realizan usando la herramienta TestCase de Django y su ejecución se realizó en línea de comando. El primer caso prueba el método de eliminación de signos de puntuación. La figura **7-1** muestra los datos de entrada y los de salida luego de la aplicación del método. Los resultados son satisfactorios.

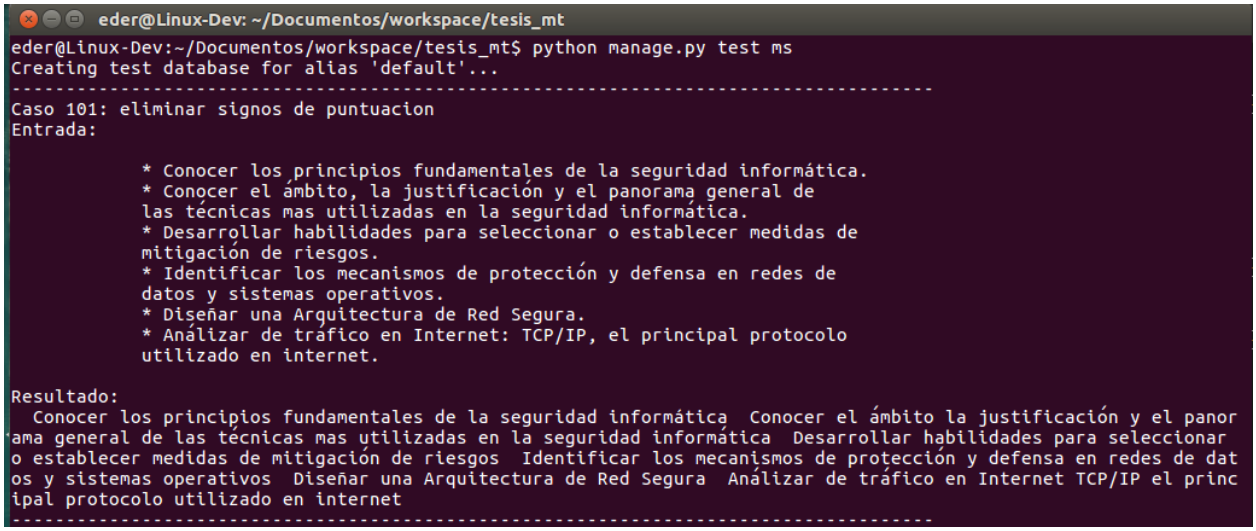

Figura 7-1: Caso 101: Eliminar signos de puntuación

El siguiente caso prueba el método de tokenización o identificación de palabras. La figura **7-2** muestra los datos de entrada y los de salida luego de la aplicación del método. Los resultados son satisfactorios.

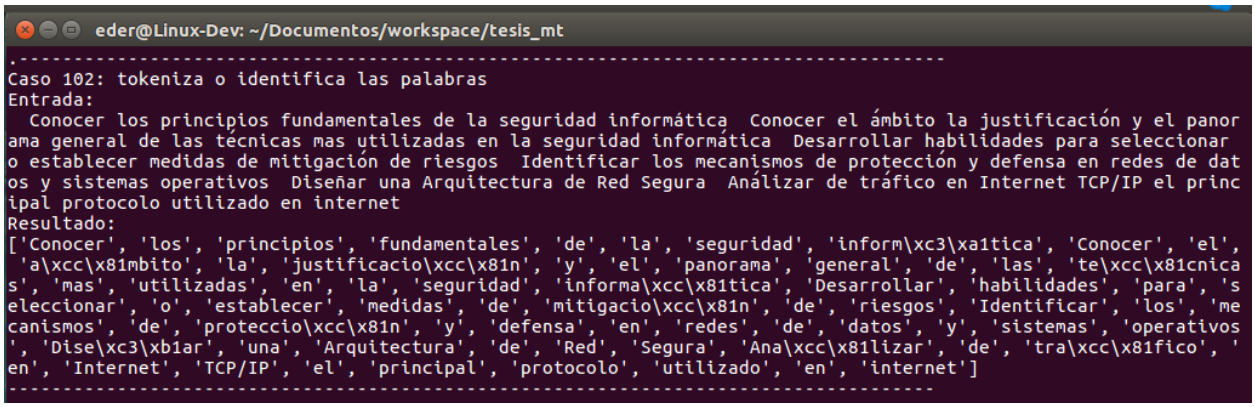

Figura 7-2: Caso 102: Tokenización de palabras

El siguiente caso prueba el método de eliminación de stopwords de un conjunto de palabras. La figura **7-3** muestra los datos de entrada y los de salida luego de la aplicación del método. Los resultados son satisfactorios.

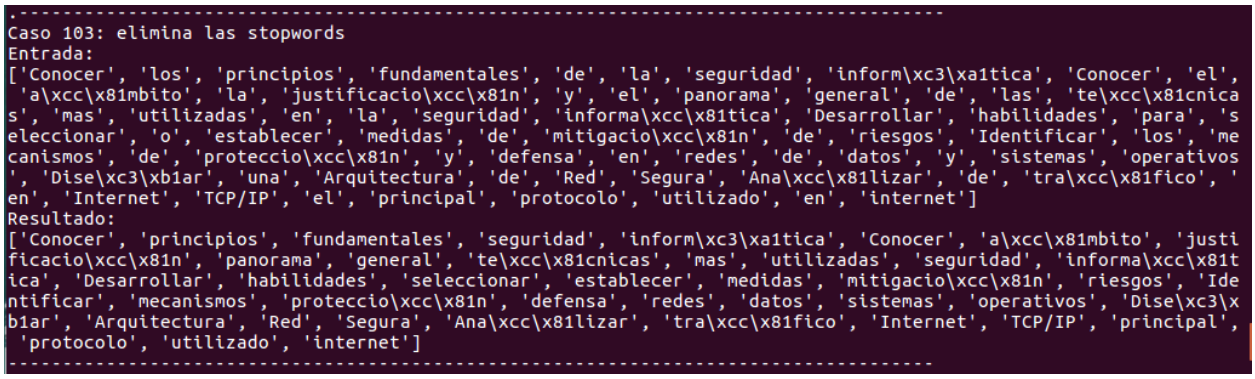

Figura 7-3: Caso 103: Eliminación de stopwords

El siguiente caso prueba el método de lematización de verbos en español a un conjunto de palabras. La figura **7-4** muestra los datos de entrada y los de salida luego de la aplicación del método. Los resultados son satisfactorios.

Figura 7-4: Caso 104: Lematizar un conjunto de palabras

El siguiente caso prueba el método de clasificación de texto según la taxonomía de Bloom. La figura **7-4** muestra los datos de entrada y los de salida luego de la aplicación del método. Los resultados son satisfactorios.

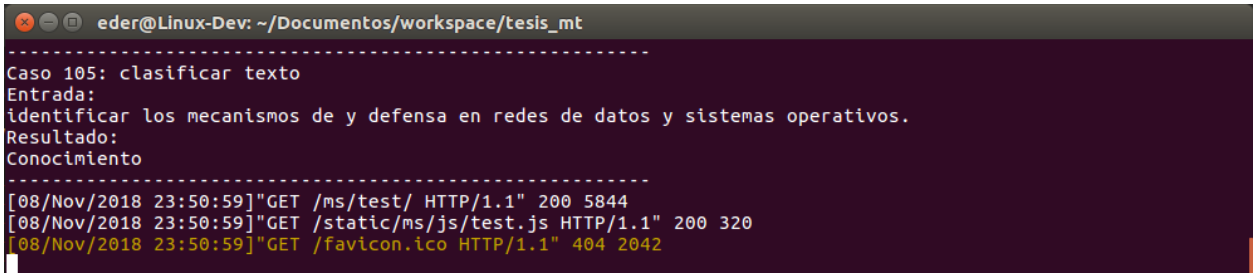

Figura 7-5: Caso 105: Clasificar texto

7.2. Pruebas de integración

El siguiente grupo de pruebas se enfoca en los formularios de la aplicación web. A este punto el aplicativo implementa el módulo de análisis de texto y se encuentra totalmente funcional. Las pruebas suponen unos datos de entrada en los formularios, luego se ejecuta la función dentro del mismo y finalmente se muestra como se afecta la base de datos. Los siguientes casos se describen textualmente, de manera que se detalla la vista afectada (a la que se hace referencia por su imagen correspondiente en el capítulo de desarrollo), los pre-requisitos necesarios, la entrada requerida y su correspondiente salida.

El siguiente caso prueba el funcionamiento de la carga individual de documentos y su pre-procesamiento. Los resultados son satisfactorios.

Caso: 201 - Guardar un nuevo documento.
Vista: Nuevo documento (ver figura **6-17**)
Pre-requisito: Debe existir al menos un registro de curso y un registro de formato de secciones para el documento.
Entrada:
Archivo Excel syllabus Gestión de la seguridad informática.
Salida:
Un corpus pre-procesado para cada una de las siguientes secciones del documento, además de un registro en base de datos para cada uno:

- Justificación
- Objetivo general
- Objetivos específicos
- Metodología
- Competencias genéricas
- Competencias del saber
- Competencias del hacer
- Competencias del ser

La figura **7-6** muestra el registro en base de datos. Se evidencia la estructura de secciones y el registro de un corpus por cada sección.
Salida esperada: Satisfactoria

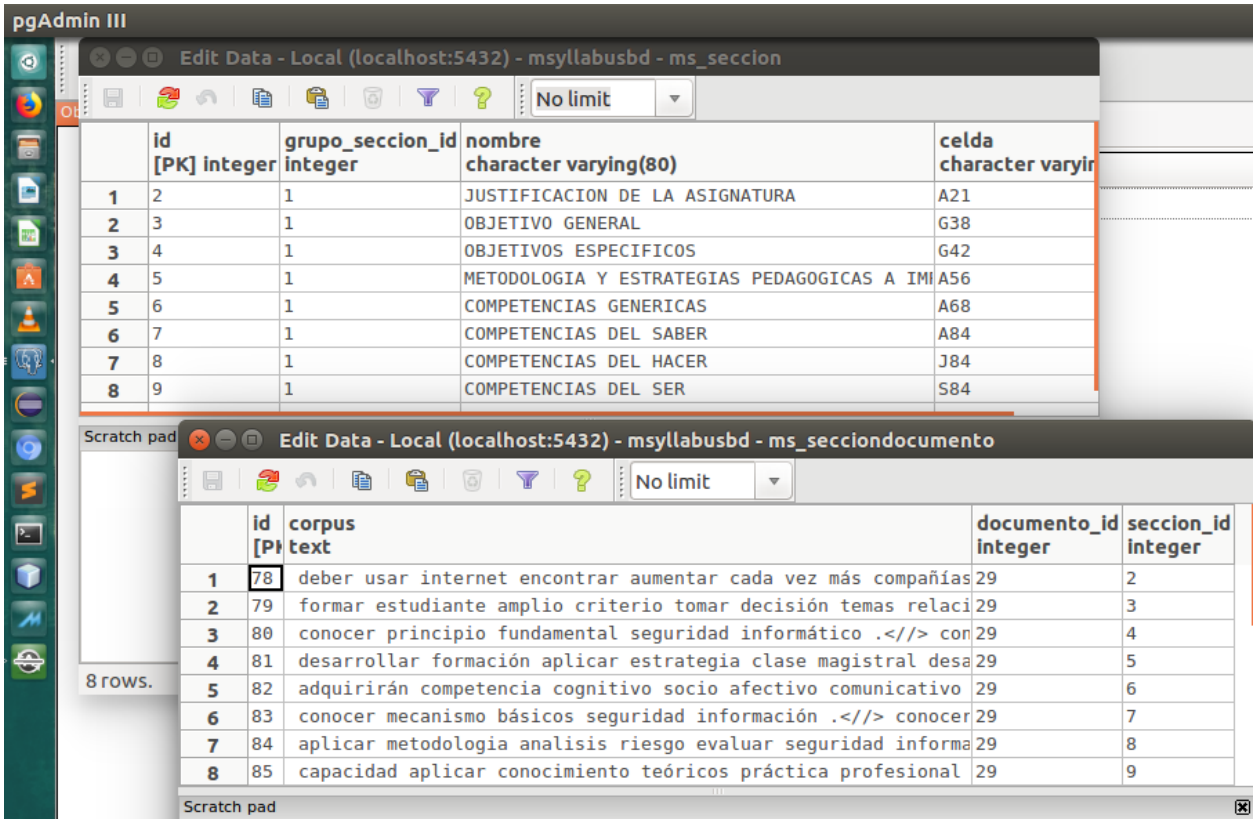

Figura 7-6: Registro de documento en base de datos

El siguiente caso prueba el análisis y clasificación de las secciones de un documento, luego que este haya sido cargado como describe el caso anterior. Los resultados son satisfactorios.

Caso: 202 - Analizar documento.
Vista: Análisis de documento (ver figura **6-18**)
Pre-requisito: Debe existir al menos un documento registrado en base de datos.
Entrada:
Salida del caso 201.
Salida:
La sección de justificación del documento contiene cuatro (4) oraciones, de las cuales dos (2) no tienen clasificación y dos se clasifican en el nivel **Aplicación** (ver figura **6-19**).

La sección de objetivos tiene seis (6) oraciones, de las cuales una (1) no tiene clasificación, una (1) tiene nivel de **Conocimiento**, tres (3) de **Aplicación** y la última de **Evaluación**.

El registro de resultados en base de datos se puede ver en la figura **7-7**.
Salida esperada: Satisfactoria

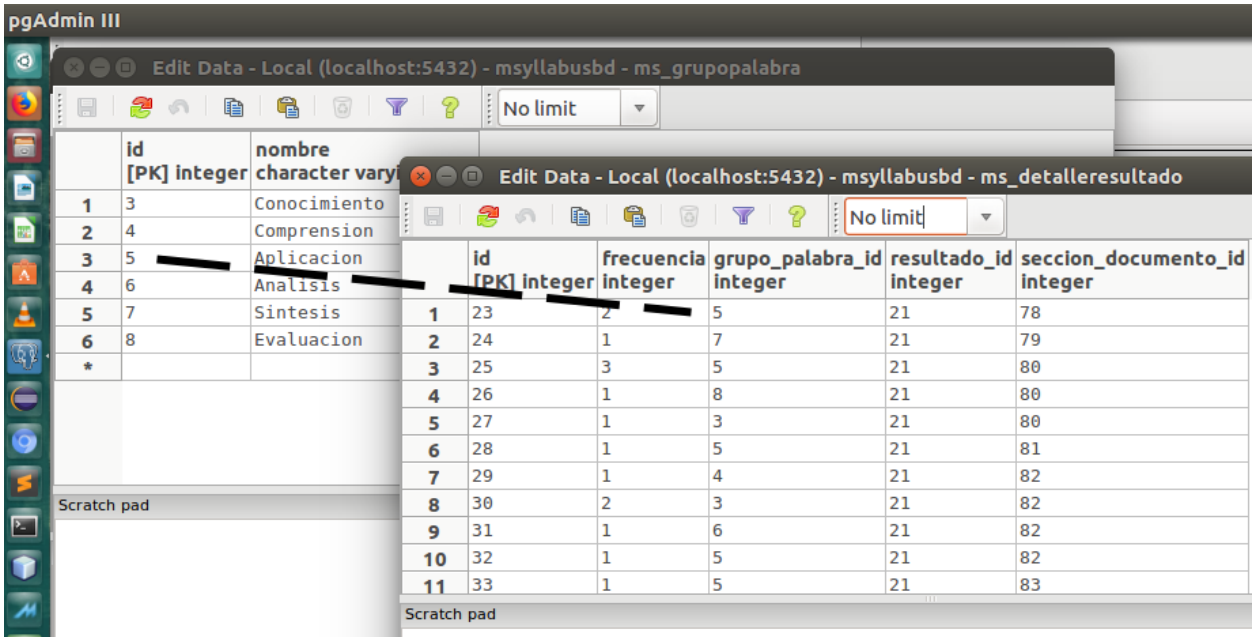

Figura 7-7: Registro de resultados en base de datos

7.3. Pruebas generales del sistema

Para determinar el grado de cumplimiento de los objetivos propuestos, fueron realizadas las pruebas correspondientes sobre la funcionalidad del software, para lo cual fue necesario realizar ciertos pasos. Posteriormente a la realización del logueo para ingresar al sistema, las pruebas consistieron en realizar la carga del documento correspondiente por la asignatura seleccionada, colocándole una descripción específica a cada uno de los documentos a analizar. Luego de esto, el sistema procede a realizar la división por secciones del documento para que el análisis sea mucho más preciso, lo cual después de realizar este, nos arrojaba la interpretación que le dio al documento, mostrando un enlace al cual se debe acceder para poder realizar una vista en general de los resultados que fueron obtenidos de la interpretación que este realizó, determinando así la clasificación que tiene cada sección en el documento.

Luego de las pruebas que fueron realizadas, fueron dadas algunas recomendaciones que surgieron en el transcurso de la puesta en marcha del sistema, entre estas se encuentran que al momento de seleccionar la asignatura a la cual se le va a cargar el documento, este herede o tome el código de la asignatura por defecto, para así evitar estar colocándolo manualmente. Junto a esto, se recibió la recomendación de que pueda realizarse la carga de documentos por lotes, ya que en la forma como este se encuentra desarrollado hay que cargar los documentos uno a uno y resulta muy tedioso debido a que existen muchas asignaturas por realizarles el estudio, agregándole a esto que pueda realizarse un análisis en general de las asignaturas para que al final los resultados puedan ser visualizados de forma numérica y así poder tener una mejor forma de

contemplar estos, teniendo además de esto la opción de realizar la exportación de los datos que fueron obtenidos por medio de un archivo en Excel para posteriores estudios que se necesiten. Por último, determinar y mostrar los porcentajes a los que equivalen los resultados mostrados en los gráficos finales, añadiéndole a esto que deben mostrarse en las leyendas, la sección a la cual pertenece ese resultado en específico que se está mostrando.

7.4. Resultados

Tomando como insumo el syllabus de la asignatura 'Gestión de la seguridad informática', se describen a continuación los resultados obtenidos. Las tablas a continuación muestran los resultados por cada sección. Cada tabla describe los niveles presentes, las frecuencias absolutas y las relativas. La medición se realiza en proporción a la cantidad de oraciones por sección. La opción *Sin clasificar* corresponde a las oraciones que, luego de ser analizadas, no tienen presencia de niveles en la taxonomía de Bloom.

Tabla 7-1: Resultado Justificación

Nivel	Frecuencia absoluta	Frecuencia relativa
Aplicación	2	50 %
Sin clasificar	2	50 %

Tabla 7-2: Resultado Objetivo general

Nivel	Frecuencia absoluta	Frecuencia relativa
Síntesis	1	100 %
Sin clasificar	0	0 %

Tabla 7-3: Resultado Objetivos específicos

Nivel	Frecuencia absoluta	Frecuencia relativa
Aplicación	3	50 %
Evaluación	1	16.67 %
Conocimiento	1	16.67 %
Sin clasificar	1	16.67 %

Tabla 7-4: Resultado Metodología

Nivel	Frecuencia absoluta	Frecuencia relativa
Aplicación	1	50 %
Sin clasificar	1	50 %

Tabla 7-5: Resultado Competencias genéricas

Nivel	Frecuencia absoluta	Frecuencia relativa
Comprensión	1	9.09 %
Conocimiento	2	18.18 %
Análisis	1	9.09 %
Aplicación	1	9.09 %
Sin clasificar	6	54.54 %

Tabla 7-6: Resultado Competencias del saber

Nivel	Frecuencia absoluta	Frecuencia relativa
Conocimiento	1	20 %
Aplicación	1	20 %
Sin clasificar	3	60 %

Tabla 7-7: Resultado Competencias del hacer

Nivel	Frecuencia absoluta	Frecuencia relativa
Evaluación	1	33.33 %
Análisis	1	33.33 %
Sin clasificar	1	33.33 %

Tabla 7-8: Resultado Competencias del ser

Nivel	Frecuencia absoluta	Frecuencia relativa
Aplicación	3	50 %
Síntesis	1	16.67 %
Sin clasificar	2	33.33 %

Para calcular las frecuencias se utiliza la oración como unidad de medida. Es decir, se divide el contenido del texto en sus respectivas oraciones. Cada una de estas es analizada y clasificada de forma independiente. La clasificación resultante identifica que nivel se hace presente en cada una de ellas. Posteriormente se agrupan los niveles resultantes y se suman los coincidentes. Las oraciones que no se clasifican dentro de ningún nivel, se suman con la etiqueta sin clasificar. Ya teniendo la frecuencia absoluta, se puede calcular la frecuencia relativa.
La tabla de frecuencias permite a los investigadores la toma de decisiones. Dependiendo de los niveles presentes en cada sección, se puede indicar si estos son adecuados o no, y con respecto a los porcentajes se puede validar que la proporción sea correcta, esté por debajo de lo esperado o, incluso, por encima. Con este análisis, se puede modificar un Syllabus agregando mas verbos dentro de un nivel o actualizar las oraciones sin clasificar. La pertinencia de qué niveles deben estar presentes y en qué porcentaje esta a cargo de los investigadores. El informe arrojado por la aplicación es solo un indicador para la toma de decisiones.

8 Conclusiones

El resultado final, luego del cumplimiento del proceso de desarrollo, es un sistema de información capaz de gestionar la carga de documento, el análisis y la minería de su contenido, con respecto a un conjunto de requisitos. Para el caso concreto, estos requisitos constituyen el dominio cognitivo de la taxonomía de Bloom. El sistema es capaz de identificar los niveles presentes de dicha taxonomía, dividiendo los resultados según las secciones del documento y finalmente genera un informe gráfico que describe los resultados. El sistema es un aplicativo Web, con soporte multiplataforma y persistencia de información sobre una base de datos postgreSQL. Para lograr esto se cumplieron varios objetivos, la recolección de datos, determinar los requisitos tecnológicos, diseñar la aplicación, desarrollar el componente de análisis de texto y la evaluación del producto. El primer objetivo *Realizar el acopio de información para determinar los requerimientos funcionales y no funcionales del sistema, mediante reuniones y entrevistas con los interesados*, produjo un listado de requisitos. Este quedó redactado usando el estándar IEEE 830 (Software Requirement Specification IEEE 830), como se muestra en el capítulo cuatro (4). Esto constituye la especificación formal de los requerimientos definitivos para el sistema.

Para el segundo objetivo *Determinar los requisitos de tecnología para cumplir los requerimientos del sistema, mediante la realización de una investigación*, se tuvo en cuenta la experiencia previa en clases de Inteligencia Artificial, se decidió adoptar Python como lenguaje de desarrollo y la librería NLTK para el proceso de minería y clasificación. Esta última se encuentra muy bien documentada lo que facilitó el desarrollo. Finalmente, para el front-end se utilizó el framework django. El cual es muy productivo, debido a que automatiza las operaciones CRUD sobre las tablas de la base de datos. esto permite enfocarse en la lógica del negocio. En resumen los requisitos escogidos son: python, NLTK, django y para la persistencia se decidió por el motor postgreSQL.
Para el tercer objetivo *Diseñar la aplicación conforme a los requerimientos obtenidos para delimitar las tareas de desarrollo, usando UML para la creación de los diagramas*, se tomó la especificación de requerimientos, plasmada en el capítulo cuatro (4), como fuente para diseñar la solución. Los diagramas resultantes se han plasmado en el capítulo cinco (5) del presente trabajo. Se crearon los diagramas de casos de uso para delimitar la aplicación. Se diseñó la persistencia de los datos con el diagrama de entidad-relación. Además, se creó un diagrama de actividades para describir el proceso general del sistema.
El cuarto objetivo *Desarrollar un componente de análisis texto, para realizar minería y clasificación de contenido conforme a un grupo de requisitos, de manera que se pueda establecer cuales de ellos están presentes en un documento dado*, se desarrolló en python, usando la librería NLTK

como se explica en el capítulo seis (6). Se creó de forma independiente. Esto facilitó sus pruebas, permite que pueda ser usado en otros proyectos y su diseño facilita su integración. Los resultados de las pruebas se pueden ver plasmadas en el capítulo siete (7).
El quinto objetivo *Desarrollar la aplicación que permita dar solución a la problemática expuesta, mediante la implementación del diseño realizado*, se realizó usando el framework django como se ha mencionado. Por su puesto se integra la librería mencionada en el párrafo anterior. En el capitulo seis (6) se pueden apreciar las vistas, tanto del front-end de administración y configuración, como también el de operación general. Se anexa a este proyecto el manual de usuario del aplicativo.
El sexto objetivo *Evaluar el producto obtenido para determinar el cumplimiento de los requerimientos, mediante el uso de técnicas de prueba de calidad de software*, se ve reflejado en el capitulo siete (7). Dentro del apartado de pruebas generales, se describe las tareas realizadas y su resultado. En cuanto a las pruebas modulares, se realizaron las necesarias para validar el funcionamiento del componente de análisis. Se probaron de forma independiente todas las funciones para preprocesar texto. Al igual que las pruebas de integración para testear la clasificación. Al final la pruebas fueron satisfactorias, garantizando un producto software de buena calidad. Esto queda corroborado mediante un listado de casos de prueba. En ellos se puede apreciar la entrada suministrada y la salida obtenida.

Para el futuro se espera que el aplicativo realice análisis semántico, permita establecer parámetros con opciones avanzadas y que esto otorgue la posibilidad de sugerir opciones para mejorar el syllabus conforme a la taxonomía de Bloom. Es decir, que la aplicación ofrezca los verbos y oraciones recomendadas, de manera que se optimice el contenido del syllabus conforme a lo que se requiere. De esta forma se facilitaría la toma de decisiones para los investigadoras y haría aún mas eficiente su labor. Conforme a esto, si el investigador considera que la recomendación es óptima, entonces de aplicará de forma inmediata, de otra manera se ingresarán los cambios considerador de manera manual.
Finalmente, el componente de minería de texto se ha desarrollado de forma independiente. Esto permite que se pueda usar en otros proyectos, incluso realizar mejoras o personalizarlo para nuevos usos. Los proyectos futuros, que involucren minería de texto, pueden partir de este componente. No se requeriría iniciar de cero. Esto constituye una ventaja, debido a que futuras investigaciones pueden enfocarse en las particularidades del negocio y acelerarían su desarrollo al tener un componente funcional ya listo.

Bibliografía

[1] D. R. Krathwohl, "A revision of bloom's taxonomy: An overview," *Theory Into Practice*, vol. 41, no. 4, pp. 212–218, 2002.

[2] "Saber Pro."

[3] F. M. R. Aldape, *Cuantificación del interés de un usuario en un tema mediante minería de texto y análisis de sentimiento*. PhD thesis, Universidad Autónoma de Nuevo León, Nuevo León, June 2013.

[4] W. Martel, D. Carranco, and D. Cevallos, "Determinación de niveles de agresividad en comentarios de la red social Facebook por medio de Minería de Texto," *1*, vol. 6, p. 7, 2015.

[5] C. Cuervo Vargas, "Diseño de una metodología para la extracción de funciones y mapas relacionales a partir de herramientas de minería de texto," Master's thesis, Pontificia Universidad Javeriana, Bogotá, Colombia, 2012.

[6] M. Calvo Torres, *Text Analytics para Procesado Semántico*. PhD thesis, Universidade de Vigo, July 2017.

[7] M. Calvo Torres, "Shiny."

[8] J. Thanaki, *Python Natural Language Processing*. Packt Publishing, 1 ed., 2017.

[9] R. L. Angell, S. K. Boyer, J. W. Cooper, R. A. Hennessy, T. Kanungo, J. T. Kreulen, D. C. Martin, J. J. Rhodes, W. S. Spangler, and H. J. R. Weintraub, "System and method for using text analytics to identify a set of related documents from a source document," Nov. 2016.

[10] M. d. C. Justicia de la Torre, *Nuevas técnicas de minería de textos: Aplicaciones*. PhD thesis, Universidad de Granada, Granada, España, 2017.

[11] A. Moreno and T. Redondo, "Text Analytics: the convergence of Big Data and Artificial Intelligence," *6*, vol. 3, pp. 57–64, 2016.

[12] M. Natarajan, "Role of Text Mining in Information Extraction and Information Management," *4*, vol. 25, pp. 31–38, July 2005.

[13] C. de la República de Colombia, "Ley número 23 de 1982," Jan. 1982.

[14] P. de la República de Colombia, "Decreto 1360 de 1989," June 1989.

[15] C. A. La comisión del acuerdo de Cartagena, "Decisión 351," Dec. 1993.

[16] F. S. Foundation, "GNU General Public License," June 2007.

[17] J. Lozada, "Investigación Aplicada: Definición, Propiedad Intelectual e Industria," *1*, vol. 3, pp. 34–39, Dec. 2014.

[18] V. Vaishnavi, B. Kuechler, and S. Petter, "Design Science Research in Information Systems," Jan. 2004.

[19] S. E. S. Committee, "IEEE Recommended Practice for Software Requirements Specifications," Oct. 1998.

[20] P. S. Foundation, "Python," 1991.

[21] G. Bonaccorso, *Machine Learning Algorithms*, vol. 1. Birmingham, UK: Packt Publishing, July 2017.

[22] D. S. Foudation, "Django project," 2005.