2016-09

Construcción de una memoria organizacional a partir de textos no estructurados usando herramientas de minería de texto

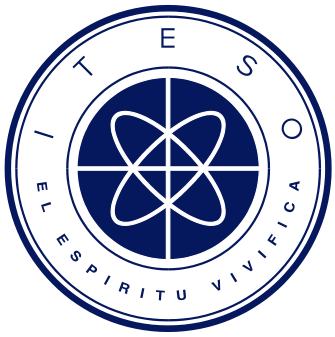
Espíritu-Sandoval, Karla Y.

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

MAESTRÍA EN SISTEMAS COMPUTACIONALES



Construcción de una memoria organizacional a partir de textos no estructurados usando herramientas de minería de texto.

Trabajo recepcional que para obtener el grado de

MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: Karla Yuriza Espíritu Sandoval.

Director: Dr. Víctor Hugo Zaldívar Carrillo
Codirector: M. en C. Rodrigo Israel Novelo Cervera

Tlaquepaque, Jalisco, Septiembre de 2016.

Tabla de Contenido

Índice de figuras 4
Índice de tablas 5
Agradecimientos 6
Dedicatorias 7
1. Introducción 8
1.1 Contexto 8
1.2 Coordinación PAP 8
1.3 Definición del problema 10
1.4 Solución propuesta 13
1.5 Justificación 14
1.6 Hipótesis 15
1.6.1 Hipótesis Científica 15
1.6.2 Hipótesis del Sistema 15
1.7 Metas 16
1.7.1 Metas Científicas 16
1.7.2 Metas del Sistema 16
2. Metodología 17
3. Marco Teórico 18
3.1 Memoria organizacional (MO) 18
3.1.1. Definición de MO 18
3.1.2. Objetivos de MO 18
3.1.3. Estructura de MO 19
3.1.4. Clasificación de MO 20
3.1.5. Ventajas de una MO 20
3.1.6. Herramientas de MO 20
3.2 Minería de Datos (MD) 21
3.2.1 Definiciones de MD 21
3.2.2 Etapas de MD 22
3.2.3 Objetivo de MD 23
3.2.4 Aplicaciones de MD 24
3.2.5 Técnicas de MD 24
3.2.6 Herramientas de minería de datos 25
3.3 Minería de Texto (MT) 27

3.3.1 Definiciones de MT 27
3.3.2 Metodología de MT 27
3.3.3 Objetivo de MT 30
3.3.4 Componentes de MT 30
3.3.5 Documentos o texto no estructurado 30
3.3.6 Ejemplos de textos analizables con MT 30
3.3.7 Técnicas de MT 30
3.3.8 Desafíos de MT 31
3.3.9 Herramientas de MT 31
3.4 Procesamiento del Lenguaje Natural (PLN) 32
3.4.1 Objetivo de PLN 32
3.4.2 Dificultad principal 33
3.4.3 Herramientas de PLN 33
3.4.4 Componentes de PLN 34
3.4.5 Aplicaciones de PLN 34
3.4.6 Recursos de PLN 34
3.5 Clustering 35
3.5.1 Definición de *clustering* 35
3.5.2 Objetivo de *clustering* 35
3.5.3 Ventajas de *clustering* 36
3.5.4 Algoritmos de *clustering* 36
3.6 Clasificación de Palabras 36
3.6.1 Etiquetas de la clasificación 37
3.7 N-Gramas 37
3.7.1 Frecuencia de N-gramas en un documento 38
3.7.2 Elementos que conforman un N-grama 38
3.8 Nube de palabras 38
3.9 Bolsa de palabras 39
4. Construcción de la memoria organizacional 40
4.1. Definición de la memoria organizacional 40
4.2. Conocer y obtener información de los proyectos PAP 40
4.3. Selección de algoritmos 41
4.4. Análisis de tecnologías 42
4.5 Implementación 44

5. Resultados 52
5.1 Resultados de la implementación 52
5.2 Resultados de la memoria organizacional 52
6. Conclusiones 53
6.1. Conclusiones 53
6.2. Trabajos futuros 53
Bibliografía 54
Anexos 56
A. Resumen de palabras clave 56
B. Diagrama de estructura de la base de datos 57
C. Requerimientos del sistema 58
D. Diagrama de casos de uso del sistema. 60
E. Código Fuente 61
E.1. Biblioteca para convertir de formato pdf a txt 61
E.2. Instrucciones para limpiar el texto 61
E.3. Tokenización 61
E.4. Crear N-Gramas 61
E.5. Clasificación de palabras 61
E.6. Identificación de competencias 64
E.7. Frecuencias 64
E.8. Nube de palabras 67

Índice de figuras

Figura 1. Proceso actual para creación y desarrollo de proyectos de aplicación profesional. 11
Figura 2. Proceso propuesto para la creación y desarrollo de PAP. 13
Figura 3. Estructura de una MO [4]. 19
Figura 4. Proceso de extracción de conocimiento en bases de datos [9]. 22
Figura 5. Etapas de MD [8]..... 23
Figura 6. Elementos del proceso de la MT [17]..... 29
Figura 7. Ejemplo de texto no estructurado. 30
Figura 8. Ejemplo de nube de palabras. 38
Figura 9. Estructura de la memoria organizacional de este caso de estudio. 40
Figura 10. Proceso para crear una memoria organizacional a partir de texto no estructurado usando herramientas de minería de texto. 44
Figura 11. Texto antes de la limpieza..... 45
Figura 12. Texto después de implementar la limpieza de texto. 45
Figura 13. Sistema PAP. 46
Figura 14. Menú de categorías. 47
Figura 15. Formulario de búsquedas. 48
Figura 16. Formulario para mostrar la nube de palabras y su frecuencia. 48
Figura 17. Formulario para mostrar grafica de frecuencia y su frecuencia. 49
Figura 18. Formulario para la consulta y creación de gráficas. 49
Figura 19. Formulario para consultar las competencias. 50
Figura 20. Reporte de competencias encontradas en el documento. 50
Figura 21. Formulario para agregar un documento a la memoria organizacional. 51

Índice de tablas

Tabla 1. Ejemplos de clasificación de palabras por categoría..... 37
Tabla 2. Ejemplo de cómo se representa la etiqueta 37
Tabla 3. Ejemplo de N-gramas con N=2, N=3..... 38
Tabla 4. Aprendizajes mencionados por estudiantes..... 41
Tabla 5. Tecnologías que se pueden usar para crear una memoria organizacional. 43

Agradecimientos

Le doy gracias a Dios por poner en mi camino a personas gentiles y humildes llenas de amor y felicidad, personas a las que por una u otra razón conocí y compartimos sueños y metas, aquellas personas que me regalaron tiempo de sus ocupadas vidas y que gracias a ellas nunca perdí el ánimo de seguir adelante.

Le doy gracias a mis profesores de la maestría, por su apoyo, paciencia y dedicación al compartir sus conocimientos y aportar a los míos.

Le doy gracias a mis asesores: Dr. Víctor Hugo Zaldívar Carrillo y M. en C. Rodrigo Israel Novelo Cervera por su paciencia, dedicación, por guiarme en los objetivos del trabajo.

Le doy gracias a la Mtra. Martha Gabriela Muñoz Padilla por su apoyo y tiempo para realización del proyecto.

Le doy gracias a mis compañeros de maestría Vero, Toris, René, Nadia y Luis por su apoyo en clase y paciencia en algunas tareas y proyectos.

Le doy gracias a mi mamá que siempre ha confiado en mí capacidad de lograr mis sueños. Así como a mi hermana que me ha regalado una de mis grandes motivaciones (mi hermosa sobrina Abril). A mi gran y pequeña familia que siempre estuvo a mi lado.

Le doy gracias a mis amigos, Nancy, Fabián, Noé y Ricardo que siempre me brindaron su apoyo y se unían a mis desvelos en la elaboración de tareas y proyectos.

Le doy gracias a mi novio Adolfo por aguantarme todo este tiempo de escuela, gracias por aguantar las quejas, llantos y frustraciones, en especial por hacerme sentir una persona valiosa a tu lado.

Le doy gracias al Conacyt, ya que fui becario con el número 304114, su apoyo fue parte importante para la realización y término de la maestría. Gracias por apoyar los sueños de estudiantes que tienen ganas de aprender y construir un mejor futuro.

Dedicatorias

Le dedico este trabajo de tesis a mi familia que gracias a ellos esto no pudo haber sido posible, le agradezco por su gran apoyo y les doy gracias por todo lo que me han enseñado, en especial por buscar y trabajar por nuestros sueños.

1. Introducción

1.1 Contexto

Hoy en día las universidades y escuelas no sólo se preocupan por que el alumno egrese con un conocimiento académico, sino que también salga de la escuela con una experiencia real de lo que enfrentará en la vida laboral. En el Instituto de Estudios Superiores de Occidente (ITESO) se implementa un sistema de Proyecto de Aplicación Profesional (PAP), en el cual, los alumnos pueden tener un caso real que les permitan poner en práctica el conocimiento adquirido en las aulas. El PAP es un requisito para que el alumno acredite la asignatura del mismo nombre, y como parte de ésta, se valida la opción terminal (antes tesis) de las licenciaturas.

1.2 Coordinación PAP

Desde el 2006, en el programa de licenciaturas del ITESO se incluyen los PAP. Este proceso formativo se da inicio con la primera formulación del proyecto universitario que define cada estudiante al inicio de la carrera y tiene como objetivo el proyecto socio-profesional que el alumno construye en su paso por la universidad [1].

A través de los PAP se logra la formación social, obtención de un conocimiento teórico profundo de las estructuras sociales, desarrollo de capacidades reflexivas y la sensibilidad mediante la interacción con problemáticas y grupos sociales a partir de casos concretos, creando conocimientos que permitan aportes sustantivos en beneficio de una sociedad más justa y generando prácticas socio-profesionales por parte de nuestros alumnos [1].

La orientación de los PAP se ofrece para dar formación para la vida a los estudiantes. Este enfoque implica un aprendizaje centrado en los estudiantes y cuyo proceso, conforme al proyecto educativo del ITESO demanda que sea:

- Significativo, para que los aprendizajes incorporados y aprendidos abonen a una integración constructiva de pensamientos y acciones, que posibiliten la apropiación de instrumentos y signos en su estructura de conocimiento de manera más permanente.
- Situado, para que los estudiantes aprendan haciendo y aplicando conocimiento junto con herramientas aprendidas en clases.
- Reflexivo, para que los estudiantes produzcan explicaciones y procesos de comprensión sobre el objeto de estudio y sus contenidos, y al mismo tiempo, sepan dar cuenta de su propio proceso de aprendizaje.
- Colaborativo, para que los estudiantes aprendan a hacer al lado de, y junto con otros.
- Transferible, para que los estudiantes desarrollen competencias aplicables para la vida.

Los PAP están planeados con los elementos propios de cada disciplina y con trayectos de mayor contacto y relación interdisciplinar, que cada estudiante podrá ampliar con la elección de asignaturas del área complementaria.

Para el logro de esta apuesta formativa y en específico del cumplimiento de: la opción terminal de estudios de Licenciatura, fueron enunciados y aprobados por el Consejo Académico

del ITESO en octubre de 2006 los siguientes criterios operativos:

- Que al finalizar el PAP se haga entrega formal a las instancias involucradas del producto correspondiente o bien, realizar el cierre parcial (si el proyecto contempla más de una etapa). De igual importancia resulta, en el caso de las organizaciones e instituciones que así lo requieran, una "carta de finiquito" que registre el cumplimiento de los compromisos de la universidad, de forma tal que los procesos y responsabilidades queden debidamente "cerrados".

- Que en los casos en que por alguna razón los proyectos no lleguen al término planeado, de igual manera se cuente con algún documento que clarifique los motivos de interrupción del proyecto.

- Que se diseñen estrategias diferenciadas para la difusión de los resultados de los proyectos, como evidencia histórica de lo realizado, del proceso y de la relevancia estratégica de la intervención institucional.

- Que los alumnos y el académico responsables cuenten con un espacio final de reflexión y evaluación de la propia experiencia, así como del proyecto en su totalidad, especialmente referidos al compromiso social logrado y/o procesos de transformación, tanto en los escenarios en los que se actuó como en el propio planteamiento de vida de los alumnos y docentes, con el fin de obtener aspectos de mejora continua para proyectos a futuro.

- Que al final de cada semestre la evaluación de los proyectos sea revisada por el coordinador de la Unidad Académica Básica (UAB), coordinador de la carrera, responsable del PAP, representantes de centros y académicos involucrados, incorporando criterios académicos y formativos a la ponderación de los objetivos planteados e intentando incorporar algún tipo de evaluación de los destinatarios de los proyectos.

- Que los resultados se presenten por escrito y se entregue copia al coordinador de carrera por parte del académico responsable del proyecto.

1.3 Definición del problema

Como se ha mencionado anteriormente, los PAP se concluyen con la entrega de un reporte que contiene la descripción de las actividades o productos que se realizaron en el proyecto. Los reportes son entregados en diferentes formatos digitales, los cuales se almacenan en un repositorio institucional en la biblioteca. Es importante mencionar que cada semestre se genera alrededor de 150 reportes lo que significa un incremento anual considerable. Por este motivo es necesaria la extracción de información de cada reporte.

Actualmente, el equipo de la Coordinación de Proyectos de Aplicación Profesional (CPAP) no cuenta con una herramienta para consultar la información de manera dinámica que le permita visualizar estadísticas y/o tendencias del contenido de los reportes. Con la ausencia de una herramienta automática de análisis de información es posible la pérdida de un historial de actividades y vínculos que se han generado entre instituciones, alumnos y asesores. La Figura 1, muestra el flujo de trabajo de los proyectos PAP.

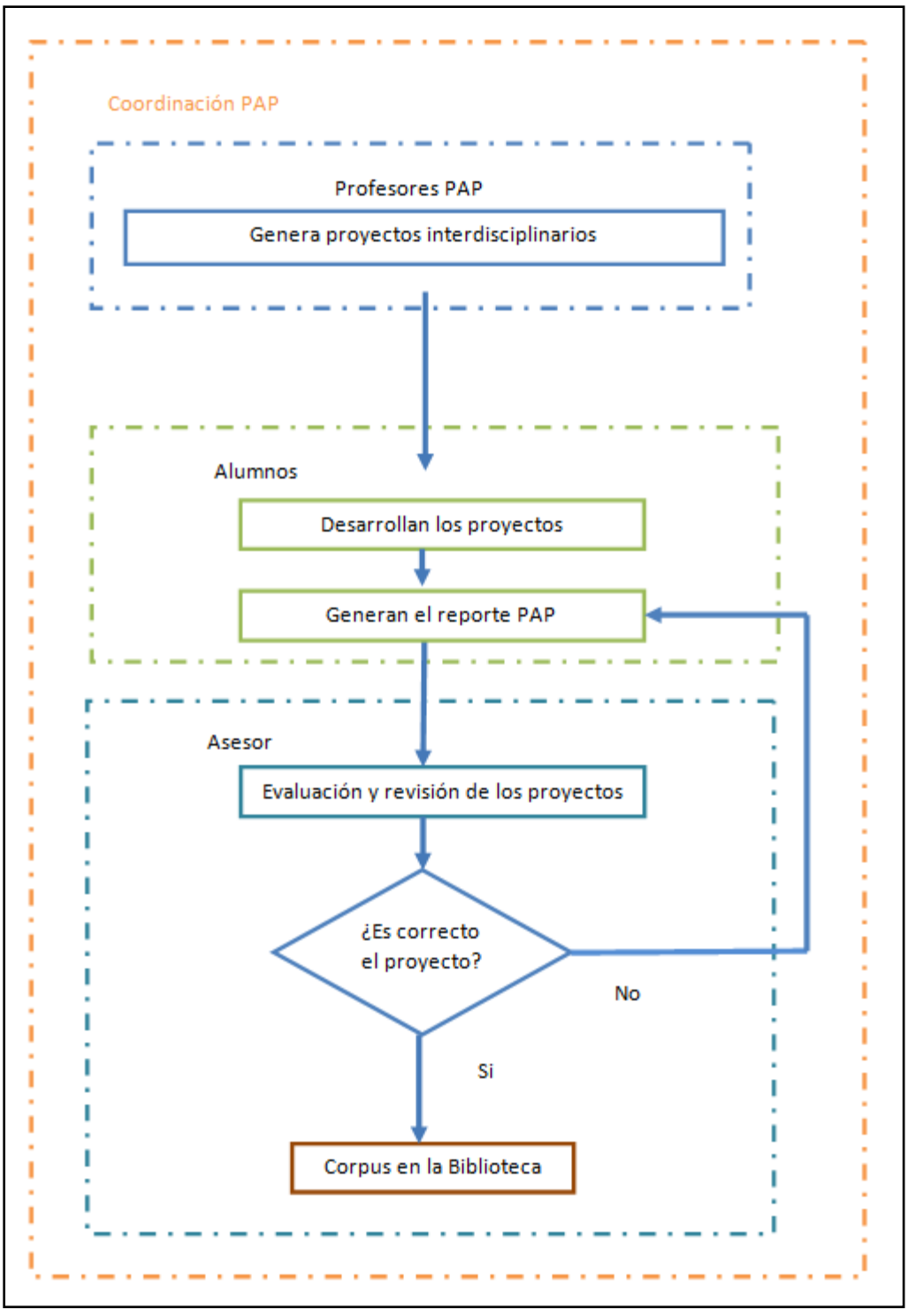


Figura 1. Proceso actual para creación y desarrollo de proyectos de aplicación profesional.

Como se observa en la Figura 1, los reportes PAP son generados por los alumnos del ITESO, en donde se describen las actividades que se generaron a partir de un proyecto que les permite aplicar los conocimientos adquiridos en el transcurso de la carrera, así como generar un conocimiento de la estructura social. Por consecuencia cada periodo (primavera, verano, otoño) se generan aproximadamente 150 reportes. Dichos reportes son evaluados por

los asesores y posteriormente almacenados en un repositorio de la biblioteca del ITESO.

La CPAP no interviene en la evaluación y análisis de los reportes PAP, por este motivo se excluye a la coordinación PAP de saber contenidos o temas que puedan enriquecer las apuestas y apoyar en generar nuevos proyectos.

1.4 Solución propuesta

Con la construcción de una memoria organizacional a partir de texto no estructurado usando herramientas de minería de texto, se brinda una herramienta a la CPAP para obtener estadísticas, información de interés, aprendizajes, experiencias plasmadas por los alumnos en los reportes PAP, logrando tener información que ayude a generar nuevos proyectos, mejorar algunos existentes y facilitar la toma de decisiones con base en la información contenida en los documentos, así como llevar un registro de las vinculaciones entre instituciones públicas y/o privadas y el ITESO. La Figura 2 muestra el flujo de trabajo propuesto.

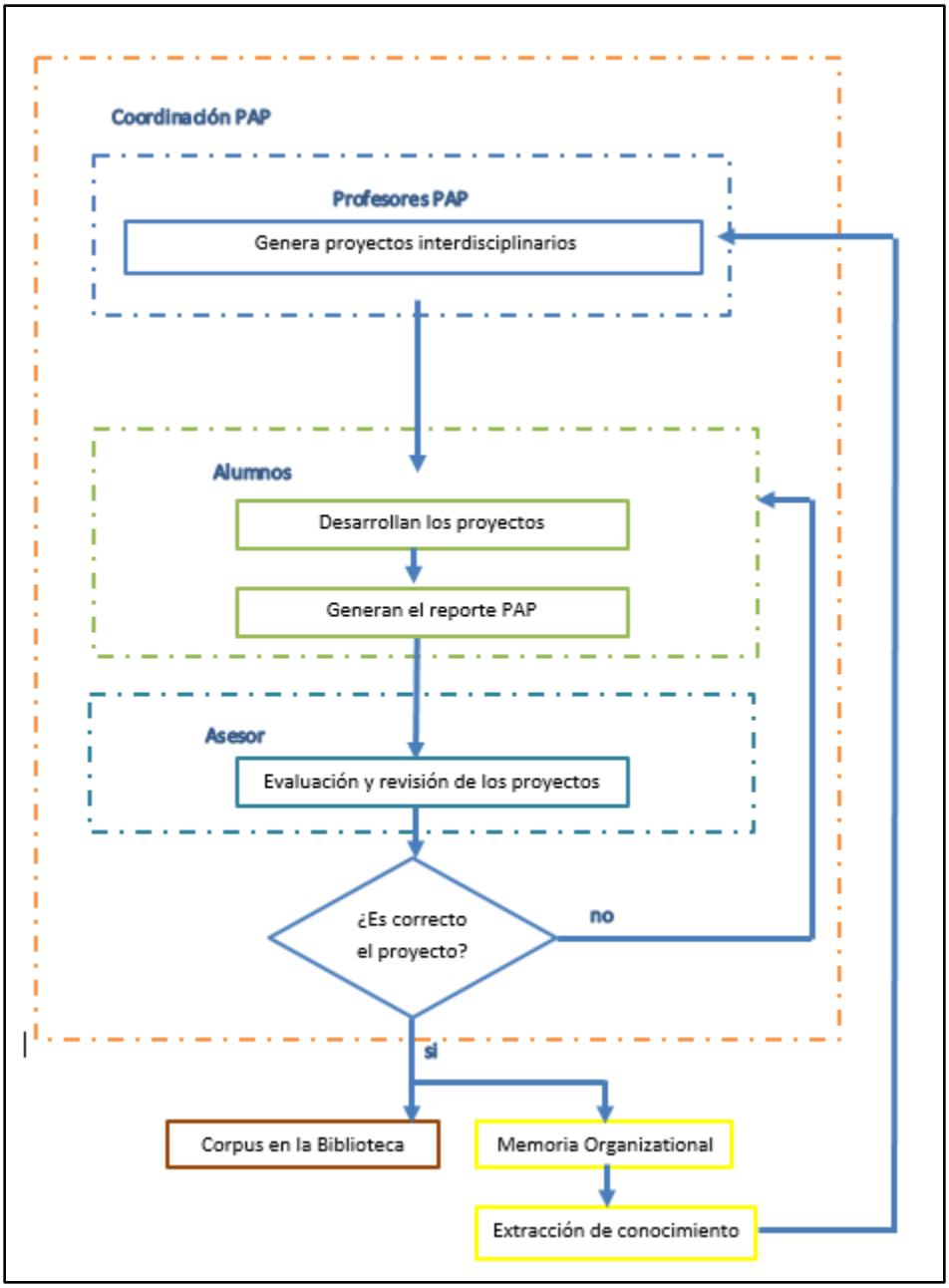


Figura 2. Proceso propuesto para la creación y desarrollo de PAP.

1.5 Justificación

Actualmente el proceso de los PAP concluye con un documento que es almacenado en la biblioteca, sin tener información de retorno a la coordinación y a los departamentos. Las actividades desarrolladas en este trabajo ayudan a recopilar información, implementar algoritmos de minería de texto que permitan analizar y generar información relevante (determinada por la CPAP), que ayuda a mejorar la comprensión de los documentos entregados por los alumnos cuando concluyen su asignatura PAP así como hacer uso de la información contenida en estos.

Para construir una memoria organizacional se parte de tecnologías y herramientas de minería de texto, Procesamiento del Lenguaje Natural (PLN) apoyadas con la minería de datos (MD). La herramienta permite fácil acceso al análisis de la información generada y almacenada en los reportes PAP.

1.6 Hipótesis

1.6.1 Hipótesis Científica

El uso de minería de texto permite la extracción de conocimiento específico basado en la información en documentos no estructurados permitiendo tener una retroalimentación de la creación de PAP.

1.6.2 Hipótesis del Sistema

Crear una memoria organizacional de los PAP y aplicar técnicas de minería de texto permite obtener información relevante para la CPAP.

1.7 Metas

1.7.1 Metas Científicas

- Aplicar conceptos y metodologías de minería de textos para la extracción de información.

- Implementar diccionarios que ayuden a la clasificación y descripción de conceptos para la realización de las búsquedas.

- Generación de un glosario de palabras relevantes a partir de un repositorio para la clasificación de documentos.

- Implementar algoritmos para una clasificación, búsquedas y extracción de información.

1.7.2 Metas del Sistema

- Construir una memoria organizacional de los PAP.
- Generar reportes con información relevante para la CPAP.
- Hacer búsquedas de palabras y frases.
- Obtener la relevancia de las competencias descritas por la CPAP.

2. Metodología

El método propuesto para construir una Memoria Organizacional a partir de texto no estructurado usando herramientas de minería de texto es la siguiente:

a) **Revisión de bibliografía sobre metodologías y herramientas**: Se revisan bibliografías acerca de las herramientas, así como conceptos, metodologías, funciones, funcionalidad de Minería de Datos (MD), Procesamiento del Lenguaje Natural(PLN), así como otras técnicas usadas para la extracción de información. Este punto fue muy interesante ya que no se tenía ningún conocimiento del tema y ese hecho al final fue muy enriquecedor para mi formación.

b) **Análisis del proceso de gestión de los reportes PAP**: Se identifican todas las etapas por las que se gestionan los reportes PAP. Este análisis ayuda a encontrar áreas de oportunidad para determinar necesidades de información y validar requerimientos de la herramienta.

c) **Análisis de algoritmos:** Se revisan algoritmos y tecnologías de minería de texto que nos ayudan a extraer e identificar información.

d) **Análisis de las plataformas:** Se analizan plataformas que permitan un desarrollo a la medida. Se usa el lenguaje de programación .net (C#) con una conexión a una base de datos (MySQL) y lenguaje Python 2.7.

e) **Implementación de los algoritmos**: Estos algoritmos están relacionados con el tratamiento de base de datos y minería de textos. Se usan bibliotecas de *Natural Lenguaje ToolKit* (NLTK por sus siglas en inglés.) para la implementación de algoritmos (*POS, Bag of Words, n-grama, frecuencias*).

f) **Validación de la información**: Se valida la información obtenida respecto a los documentos PAP entregados para el análisis. Esta actividad se realiza con la ayuda de la CPAP (nos apoya Gabriela Muñoz), ella conoce el contexto en donde se aplican los PAP ya que por ahora no se cuenta con ninguna herramienta para el análisis de los reportes PAP y no se tiene un punto de partida.

g) **Validación de resultados:** Los resultados son validados por los integrantes de la coordinación PAP.

3. Marco Teórico

El objetivo de esta sección es describir los diferentes conceptos que tienen relación con los procesos y herramientas que se utilizan para la elaboración de esta tesis, así como para la implementación y construcción de una memoria organizacional [2].

3.1 Memoria organizacional (MO)

En esta sección se habla de la memoria organizacional (MO), sus definiciones, etapas, objetivos, aplicaciones, técnicas y algunas de las herramientas que hoy en día se usan.

3.1.1. Definición de MO

Una memoria organizacional, es una representación explícita y persistente del conocimiento e información de una organización. La finalidad de una MO es facilitar a los miembros de las organizaciones el acceso y la reutilización del conocimiento necesario para la realización de tareas y la resolución de problemas [3].

Se puede conceptualizar a la memoria organizacional como el lugar donde se almacena el conocimiento organizacional generado en el pasado para utilizarlo de forma racional en el presente y en el futuro, con la característica de que este repositorio tenga fácil acceso por todos los miembros de la organización [4].

El contenido de la MO va desde lo documentado (información general de la empresa, reportes de compras, inventarios, políticas de contratación de la empresa, manuales de procedimientos, archivos de computadora) hasta lo no documentado (experiencias, formas de pensar, actitudes sobre la toma de alguna decisión, opiniones, anécdotas) y que forman el acervo cultural, conocimientos y experiencias de sus miembros [4].

3.1.2. Objetivos de MO

La construcción de una MO tiene los siguientes objetivos [3]:

- Evitar la pérdida de la información y el conocimiento generado por los autores.
- Aprovechar las experiencias obtenidas en proyectos realizados.
- Mejorar el flujo y comunicación de la información en la organización.
- Incrementar el aprendizaje de los trabajadores.
- Integrar los diferentes tipos de conocimientos existentes en la organización.

3.1.3. Estructura de MO

La Figura 3 muestra la estructura de MO:

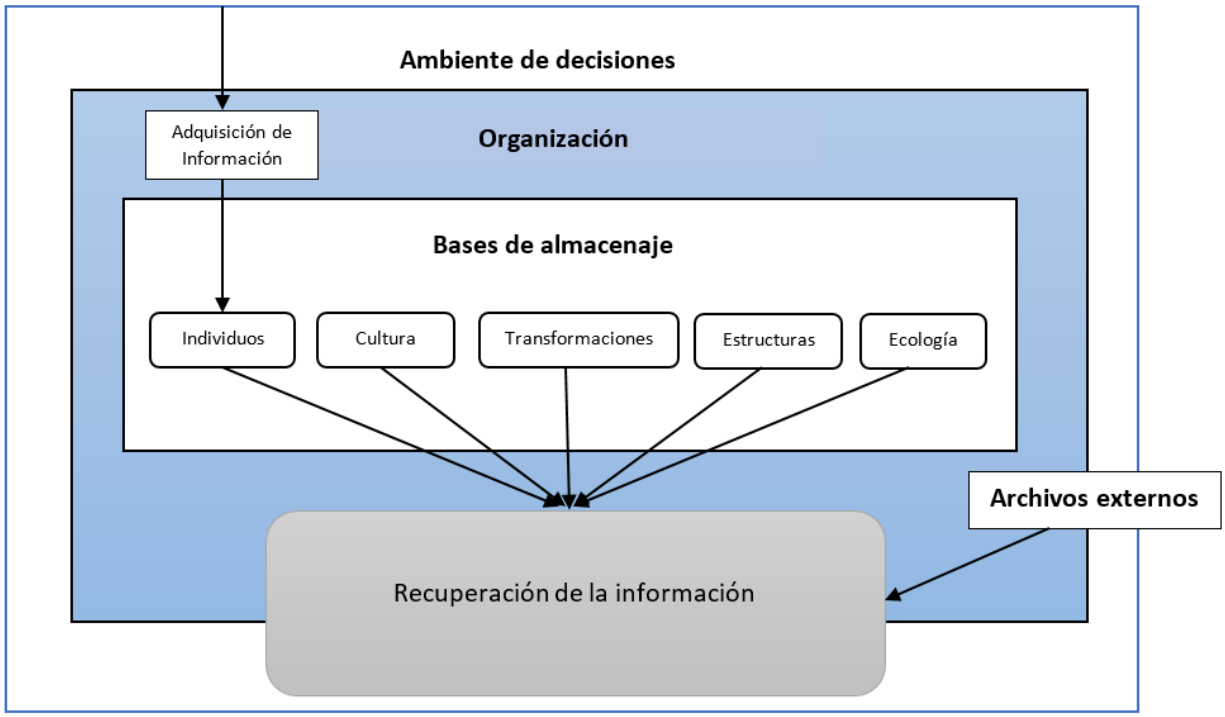


Figura 3. Estructura de una MO [4].

- **Bases de almacenaje**: Es la sección donde se puntualizan los escenarios que generan información dentro de una organización.
- **Individuos**: La información que es generada es basada en su propia experiencia y observaciones.
- **Cultura**: La información que es generada es basada en la manera que perciben, piensan y sienten los problemas transmitidos por los miembros de la organización.
- **Transformaciones**: La información se genera mediante las transformaciones que se producen en las organizaciones, es decir, la lógica que guía la transformación de una entrada (si se trata de una materia prima, un nuevo recluta, o una reclamación de seguro) en una salida (ya sea un producto terminado, un veterano de la compañía, o un pago del seguro) se materializa en estas transformaciones.
- **Estructuras**: La información es generada de los diferentes roles que tiene la organización.
- **Ecología**: La información es generada mediante la iteración con el entorno físico en el que se desarrolla, por ejemplo, lugares con poca iluminación, lugares aislados, etc.
- **Archivos externos**: La información es generada por medio de otras organizaciones, otros procesos externos a la organización.

3.1.4. Clasificación de MO

La clasificación de la memoria organizacional se da dependiendo el conocimiento contenido en ellas [6]:

- **Memoria profesional**: Compuesta por los métodos, técnicas y documentos de referencia usados en una profesión dada.
- **Memoria de gestión**: Relacionada a la organización, sus actividades, sus productos. Captura las estructuras organizacionales pasadas y presentes (recursos humanos, gestión, etc.). Esta memoria está extremadamente cercana al modelado organizacional.
- **Memoria individual**: Caracterizada por habilidades, competencias, conocimiento laboral (*know-how*) y actividades de un miembro dado de la organización.

3.1.5. Ventajas de una MO

La construcción de una MO tiene las siguientes ventajas [4]:

- Ayuda a directivos a mantener la dirección estratégica.
- Ayuda a la organización a aprovechar soluciones pasadas para atacar nuevos problemas, ya que nadie puede recordar lo que fue hecho por otros.
- El nuevo conocimiento generado por los individuos de la empresa se puede almacenar para uso posterior.
- Facilita el aprendizaje organizacional.
- Provee la facilidad de obtener la visión de expertos que estuvieron en la empresa.

3.1.6. Herramientas de MO

La MO puede ser soportada por la tecnología haciendo que el conocimiento sea recuperable y accesible [7]. Algunas herramientas que ayudan a la construcción de una memoria organizacional son las siguientes [2]:

- Intranets
- Boletines electrónicos
- Páginas amarillas
- Minería de datos
- Moodle

3.2 Minería de Datos (MD)

En este capítulo se aborda el tema de minería de datos (MD), así como su definición, objetivos, herramientas, metodologías, entre otros atributos importantes que posee.

3.2.1 Definiciones de MD

Hace varios años, la minería de datos o *Data Mining* (DM por sus siglas en inglés) apareció como una herramienta para ayudar a la comprensión de los contenidos almacenados en las bases de datos, siendo definida por algunos como "*la integración de un conjunto de áreas que tienen como propósito la identificación de un conocimiento obtenido a partir de las bases de datos que aporten un sesgo hacia la toma de decisión*" [8].

Cabe resaltar que dicha información es desconocida y permite la generación de nuevo conocimiento, que resulta de suma importancia para un proceso llamado "Extracción de Conocimiento en Bases de Datos" o *Knowledge Discovery in Databases* (KDD por sus siglas en inglés). KDD se define como "*el proceso no trivial de identificación en los datos de patrones válidos, nuevos, potencialmente útiles, y finalmente comprensibles*" [8].

Este punto es importante destacar ya que desde el punto de vista académico la minería de datos es una etapa en la KDD. KDD es el proceso de extracción de información, además se encarga de la preparación de los datos y de la interpretación de los resultados obtenidos.

El objetivo de KDD es interpretar grandes cantidades de datos y encontrar relaciones o patrones. Para conseguirlo se utilizan técnicas de aprendizaje automático (*machine learning*), estadística, bases de datos, técnicas de representación del conocimiento, razonamiento basado en casos de razonamiento aproximado, adquisición de conocimiento, redes neuronales y visualización de datos. Las tareas más comunes en KDD son la inducción de reglas, los problemas de clasificación y *clustering*, el reconocimiento de patrones, el modelado predictivo, la detección de dependencias, entre otras [9].

En la Figura 4 se observa el proceso de KDD y cómo la minería de datos está incluida en una de sus etapas:

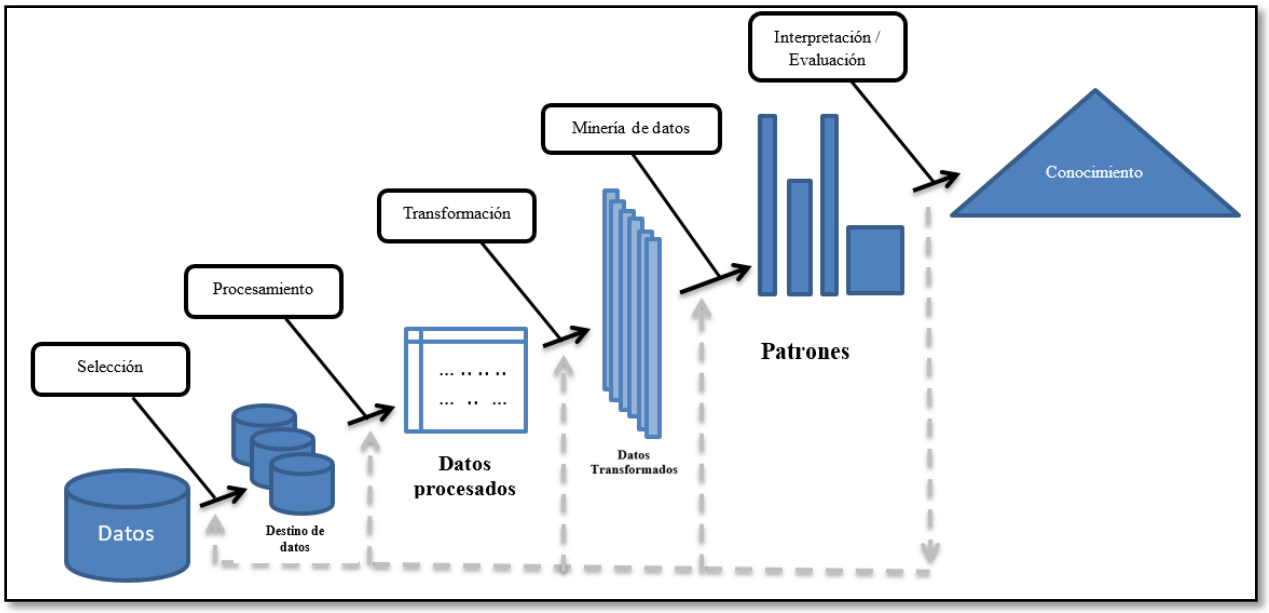


Figura 4. Proceso de extracción de conocimiento en bases de datos [9].

Otras definiciones de minería de datos:

- Proceso de extracción y refinamiento de conocimiento útil desde grandes bases de datos [9].

- Proceso de extracción de información previamente desconocida, válida y procesable desde grandes bases de datos para luego ser utilizada en la toma de decisiones [8] [10].

- Es la exploración y análisis, a través de medios automáticos y semiautomáticos, de grandes cantidades de datos con el fin de descubrir patrones y reglas significativos [9].

- Es el proceso de planteamiento de distintas consultas y extracción de información útil, patrones y tendencias previamente desconocidas desde grandes cantidades de datos posiblemente almacenados en bases de datos [9].

A partir de las definiciones mencionadas se considera para términos de esta tesis que *"la minería de datos es el proceso de extracción y refinamiento de grandes cantidades de información dando como resultado nuevo conocimiento para toma de decisiones".*

3.2.2 Etapas de MD

Dentro de la MD están establecidas cuatro etapas que se encargan de llevar un mejor análisis y diseño de la misma. A continuación, en la Figura 5, se muestran y describen las etapas de la minería de datos.

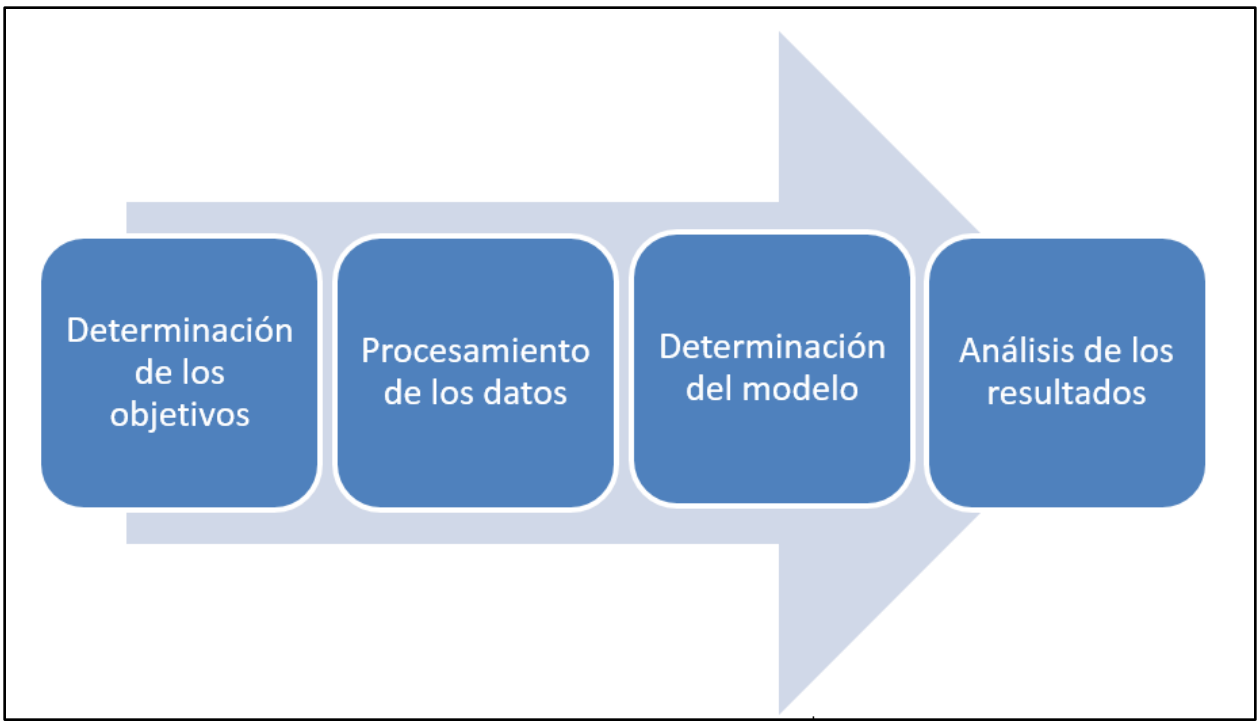


Figura 5. Etapas de MD [8].

Determinación de los objetivos: Delimitación de los objetivos que el cliente desea bajo la orientación del especialista en minería de datos.

Procesamiento de datos: Selección, limpieza, enriquecimiento, reducción y transformación de la base de datos.

Determinación del modelo: Se comienza realizando análisis estadísticos de los datos, y después se lleva a cabo una visualización gráfica de los mismos para tener una primera aproximación. Según los objetivos planeados y la tarea que debe llevarse cabo, pueden utilizarse algoritmos desarrollados en diferentes áreas de la inteligencia artificial.

Análisis de los resultados: Verificar si los resultados obtenidos son coherentes y los coteja con los obtenidos por los análisis estadísticos y la visualización gráfica. El cliente determina si son novedosos y si le aportan un nuevo conocimiento que le permita analizar mejor sus decisiones.

3.2.3 Objetivo de MD

El objetivo general del proceso de minería de datos consiste en descubrir, extraer y almacenar información relevante de grandes bases de datos de manera automática o semiautomática, combinando un conjunto de técnicas y tecnologías a partir de datos originales, estos pueden ser estructurados o no estructurados.

3.2.4 Aplicaciones de MD

A través de los años la minería de datos ha conseguido estar presente en diferentes áreas como [9]:

- **Mercadotecnia:** Identificar patrones de compra de los clientes. Determinar cómo compran a partir de sus principales características, conocer el grado de interés sobre tipos de productos, si compran determinados productos en determinados momentos.

- **Investigación Científica:** En el campo médico se almacenan grandes cantidades de información sobre los pacientes tales como: enfermedades, tratamientos impuestos, pruebas realizadas, experimentos, resultados, entre otros. Las técnicas de minería de datos pueden ayudar en esta rama para identificar terapias médicas satisfactorias para diferentes enfermedades, asociar síntomas y clasificación diferencial de patologías, identificación de terapias médicas y tratamientos erróneos para determinadas enfermedades, segmentar pacientes para una atención más inteligente según su grupo, análisis de rendimientos de campañas de información, prevención, sustitución de fármacos, entre otros.

- **Compañías de Seguros:** En el sector de las compañías de seguros y la salud privada, se pueden emplear las técnicas de Minería de Datos, por ejemplo para análisis de procedimientos médicos solicitados conjuntamente, predecir qué clientes compran nuevas pólizas, identificar patrones de comportamiento para clientes con riesgo, identificar comportamiento fraudulento, etc.

- **Telecomunicaciones:** En el sector de las telecomunicaciones se puede almacenar información interesante sobre las llamadas realizadas, tal como el destino, la duración, la fecha en que se realiza la llamada por ejemplo: detección de fraude telefónico mediante agrupamiento, se pueden detectar patrones en los datos que permitan detectar fraudes.

Las aplicaciones mencionadas en los párrafos anteriores son un pequeño grupo, ya que existe un gran universo donde la minería de datos puede ser aplicada, por ejemplo: inteligencia artificial, computación gráfica, soporte al diseño de bases de datos, procesamiento masivo, bases de datos, entre otras.

3.2.5 Técnicas de MD

Algunas de las técnicas estadísticas más conocidas para la minería de datos son [11]:

- **Redes neuronales**: Son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Algunos ejemplos de red neuronal son:
    - El perceptrón.
    - El perceptrón multicapa.
    - Los mapas auto-organizados, también conocidos como redes de Kohonen.

- **Regresión lineal:** Es la más utilizada para formar relaciones entre datos, rápida y eficaz pero insuficiente en espacios multidimensionales donde puedan relacionarse más de 2 variables.

- **Árboles de decisión:** Un árbol de decisión es un modelo de predicción utilizado en el ámbito de la inteligencia artificial. Dada una base de datos se construyen estos diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva, para la resolución de un problema. Ejemplos:
    - Algoritmo ID3.
    - Algoritmo C4.5.

- **Modelos Estadísticos**: Es una expresión simbólica en forma de igualdad o ecuación que se emplea en todos los diseños experimentales y en la regresión para indicar los diferentes factores que modifican la variable de respuesta.

- **Agrupamiento o *Clustering***: Es un procedimiento de agrupación de una serie de vectores según criterios habitualmente de distancia; se tratará de disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características comunes. Ejemplos:
    - Algoritmo K-means.
    - Algoritmo K-medoids.

- **Reglas de asociación:** Se utilizan para descubrir hechos que ocurren en común dentro de un determinado conjunto de datos.

3.2.6 Herramientas de minería de datos

Existen varias herramientas o software que permiten modelar y aplicar un diseño de la minería de datos, algunas son de uso libre y otras son soluciones comerciales, por ejemplo:

- **Dlife / Apara:** Plataforma bioinformática para la toma de decisiones clínicas. Proporciona un soporte computacional a la toma de decisiones médicas en los procesos de diagnóstico, tratamiento y seguimiento de la evolución de los pacientes que permite a los profesionales clínicos incrementar su precisión y la eficiencia de la prestación sanitaria en una media del 20% [12].

- **Clementine / SPSS:** Permite desarrollar modelos predictivos y desplegarlos para mejorar la toma de decisiones. Está diseñada teniendo en cuenta a los usuarios empresariales, de manera que no es preciso ser un experto en minería de datos [12].

- **Weka:** Colección de algoritmos de aprendizaje automático para tareas de minería de datos. Los algoritmos bien se pueden aplicar directamente a un conjunto de datos o llamadas desde su propio código de Java [13].

- **SAS Analytics / SAS:** Suite de soluciones analíticas que permiten transformar todos los datos de la organización en conocimiento, reduciendo la incertidumbre, realizando predicciones fiables y optimizando el desempeño [12].

Una extensión de minería de datos es la Minería de Texto o *Text Mining* (TM, por sus siglas en inglés) que aborda conceptos, aplicaciones, técnicas y métodos que usa para poder extraer patrones o conocimientos no textuales en documentos no estructurados.

3.3 Minería de Texto (MT)

En esta sección veremos información respecto a minería de texto (MT), definiciones, metodologías, objetivos, entre otros conceptos.

3.3.1 Definiciones de MT

Minería de texto es una extensión de la MD. La MT se enfoca en los documentos y a partir de ellos extrae información. A continuación, son mencionadas algunas definiciones de la MT:

- Un proceso de análisis exploratorio de datos que lleva información desconocida, o respuestas para las preguntas que no se tiene información estructurada [14].

- La MT se enfoca al descubrimiento de conocimiento a partir de bases de datos textuales [15]. Se refiere en general al proceso de extracción de patrones o conocimientos interesantes y no triviales de documentos de texto no estructurado [16].

3.3.2 Metodología de MT

La MT cuenta con una metodología propia para llevar a cabo el proceso de extracción de información. Esta metodología puede ser entendida como el proceso mediante el cual se llevan a cabo una serie de tareas ordenadas orientadas a la consecución de tres objetivos principales [17]:

- **Establecimiento del corpus**: El propósito principal del establecimiento del corpus es la recolección de todos los documentos relacionados con el contexto a estudiar. La colección resultante puede estar compuesta por documentos de texto, XML, emails o páginas web, entre otros. Una vez terminado el proceso de recolección, los documentos se organizan y transforman de tal forma que al final todos estén en el mismo formato (por ejemplo, en ficheros de texto ASCII) y puedan ser procesados por una máquina. La organización puede ser sencilla y consistir en una colección almacenada en un directorio o más elaborada y consistir en una lista de enlaces a páginas web de un dominio específico [17].

- **Creación de la matriz de términos**: En este paso, se utiliza el corpus para crear la matriz de términos del documento, también conocida como *term-document matrix* (TDM, por sus siglas en ingles). En dicha matriz, las filas representan los documentos, mientras que las columnas representan los términos. La relación entre términos y documentos se expresa mediante índices, es decir, medidas relacionales que pueden ser tan simples como el número de ocurrencias de cada término en cada documento. La meta consiste en representar la esencia del corpus a través de la matriz de términos. Como sabemos, no todos los términos presentes en un documento lo caracterizan: los artículos, verbos auxiliares, signos de puntuación, pronombres, etc. no tienen poder de diferenciación y por tanto no deben formar parte de la matriz de términos. A esta lista de términos que no tiene poder de diferenciación se la conoce con el nombre de *stopwords*. Además de tener cuidado con las *stopwords*, otra de las filtraciones que se pueden llevar a cabo es aquella conocida como *stemming*. El *stemming* consiste en reducir un término a su raíz para que diferentes formas gramaticales o declinaciones verbales se identifiquen con un mismo término. La construcción inicial de una matriz de términos debe incluir todos aquellos términos identificados en el corpus (columnas) a excepción de aquellos presentes en la

lista de *stopwords*, todos los documentos del corpus (filas) y la ocurrencia de cada término en cada documento (intersección fila-columna o celda) [17].

Una vez que se haya calculado la frecuencia inicial de los términos se pueden aplicar transformaciones adicionales. Generalmente, el número de apariciones de un término dentro de un documento viene a decir la importancia que tiene en dicho documento, pero no es razonable pensar que la frecuencia tiene igual importancia en un documento u otro. Por ejemplo, si un término aparece una vez en el documento A y tres veces en el documento B no es razonable concluir que ese término es tres veces más importante como descriptor del documento B que como descriptor del documento A. Para construir adecuadamente los índices existen los siguientes procesos de normalización:

Frecuencias logarítmicas: Esta transformación disminuye el impacto de las frecuencias originales (*raw frequencies*) y cómo afectan al resultado de los análisis y cómputos [17].

Frecuencias binarias: La matriz de términos resultante sólo contendrá 1s y 0s para indicar la presencia o ausencia de las respectivas palabras o términos. Esta transformación disminuye el impacto de las frecuencias originales en los análisis y cómputos que se realicen.

Frecuencias inversas (*Inverse Document Frequencies*): Consiste en una transformación muy útil que refleja tanto la especificidad del término (frecuencia de documentos) como la frecuencia total de aparición (frecuencia de términos) [17].

- **Extracción del conocimiento**: Tras la construcción de la matriz de términos, se extraen patrones en el contexto del problema específico que está siendo estudiado. Las principales categorías en las que se cataloga la extracción de la información son tres: clasificación, asociación y análisis de tendencias.

  Clasificación: La clasificación, como su propio nombre indica, consiste en clasificar los datos en un conjunto de categorías o clases. En el contexto de la minería de textos se conoce como "clasificación de textos". Un ejemplo de esto es que dado un conjunto de temas y una colección de documentos de texto se tenga que encontrar cuál es el tema de cada documento a través de modelos construidos mediante conjuntos de datos de prueba. En la actualidad, la clasificación de textos automática se aplica a filtros de spam, clasificación de páginas web, generación automática de metadatos, detección de género y muchos otros. Los dos enfoques principales encargados del estudio de la clasificación de textos son la Ingeniería del Conocimiento y el Aprendizaje Automático.

  Asociación: La idea que subyace bajo la asociación es generar reglas que identifiquen los conjuntos de frecuencias que van juntos. En la minería de textos, la asociación hace referencia a las relaciones directas entre términos o conjuntos de conceptos. La minería de textos utiliza reglas de asociación para analizar la literatura

que se ha publicado (noticias y artículos académicos publicados en la web), entre otros. El propósito principal es identificar automáticamente las asociaciones entre diferentes conjuntos.

Análisis de tendencia: Los métodos más recientes de análisis de tendencias se basan en la noción de que varios tipos de distribuciones de términos son funciones de colecciones de documentos, o lo que viene a ser lo mismo, diferentes colecciones llevan a diferentes distribuciones de términos para el mismo conjunto de términos. Por tanto, es posible comparar dos distribuciones que son idénticas a excepción de su procedencia

Los elementos que intervienen en el proceso de la minería de textos se pueden observar en la Figura 6:

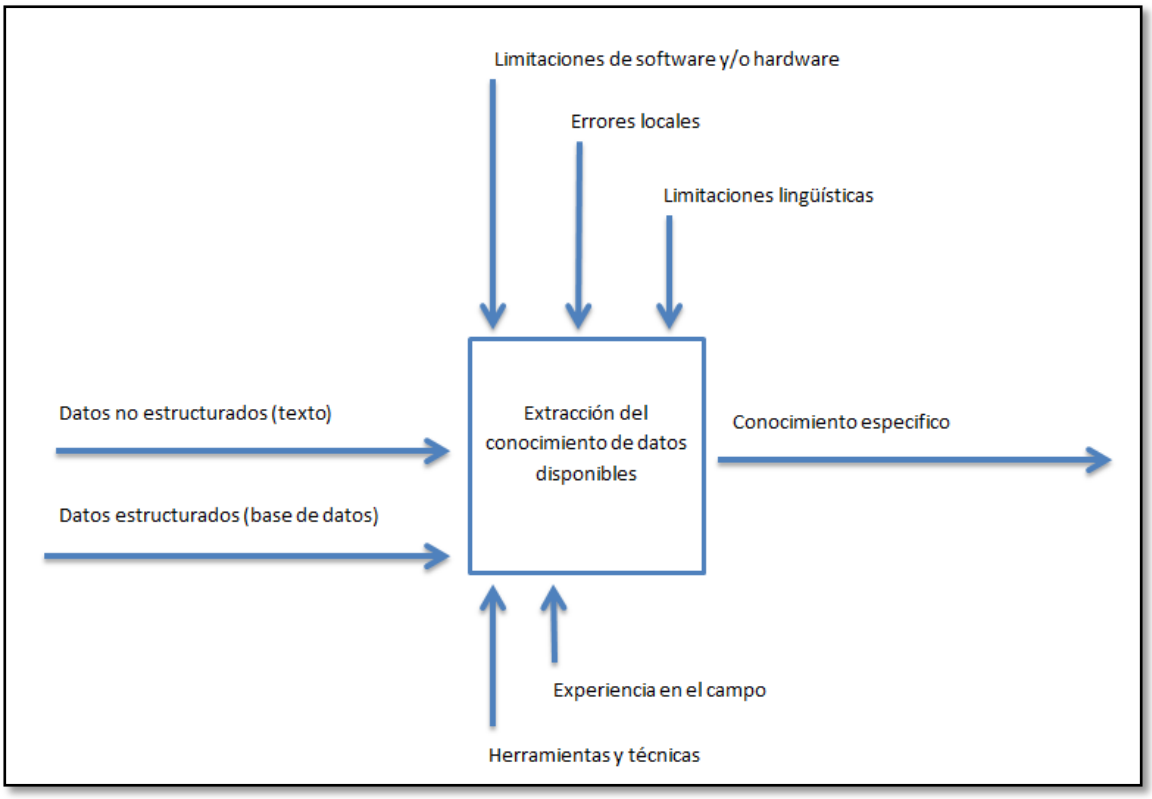


Figura 6. Elementos del proceso de la MT [17].

En la Figura 7 se identifica las entradas, que vienen a ser la recopilación de los datos estructurados y no estructurados. También se identifica la salida, es decir, el conocimiento extraído de las entradas por medio de la MT. Así mismo, se ve que existen una serie de restricciones, encabezadas por las limitaciones impuestas por el *software* o el *hardware*, cuestiones de privacidad y la dificultad que entraña el procesamiento de un texto escrito en lenguaje natural, es decir, en el lenguaje que hablan los humanos y no las máquinas. Por otro lado, también intervienen las herramientas, mecanismos y técnicas que se utilizan de forma específica durante el proceso.

3.3.3 Objetivo de MT

En la MT, el objetivo es descubrir conocimiento hasta ahora desconocido, algo que nadie sabe aún y así no podría haber sido escrito todavía, incluyendo descubrimiento de patrones y tendencias en los datos, asociaciones entre entidades, reglas de predicción, entre otros [7].

3.3.4 Componentes de MT

La MT comprende tres actividades fundamentales [18]:

- Recuperación de información seleccionando los textos pertinentes.
- Extracción de la información incluida en esos textos (hechos, acontecimientos, datos clave, relaciones entre ellos, etc.).
- Encontrar asociaciones entre esos datos claves previamente extraídos de entre los textos.

3.3.5 Documentos o texto no estructurado.

La MT trabaja con textos no estructurados, este tipo de texto no posee una estructura definida, es decir, es texto libre de formato. Ninguna parte del contenido tiene más importancia que otra. De ahí que la extracción de información en este tipo de textos no es una tarea sencilla y generalmente implica una etapa de pre-procesamiento. El texto no estructurado consiste generalmente de prosa en forma natural (i.e. documentos de texto creados con algún procesador de textos, el cuerpo de un mensaje de correo electrónico, etc.). La Figura 7 muestra un ejemplo de texto no estructurado [19].

Los felinos son mamíferos cazadores sigilosos. Entre los felinos más conocidos se encuentran: el jaguar, el tigre, el puma, etc.

Los insectos son artrópodos caracterizados por tener antenas. Por ejemplo, una hormiga es un insecto con antenas acotadas y patas largas. Una abeja es otro ejemplo de insecto.

Figura 7. Ejemplo de texto no estructurado.

3.3.6 Ejemplos de textos analizables con MT

Es fácil encontrar ejemplos de textos donde la MT puede ser aplicada, algunos ejemplos como registros médicos, reclamos de garantías, registros de *call centers*, *memorándums*, notas, encuestas de texto libre, páginas web, declaraciones aduaneras o impositivas, retroalimentación de clientes, mensajes de correo electrónico, reclamos de seguros, información de patentes, alertas, etc. [20].

3.3.7 Técnicas de MT

Actualmente un promedio del 80% de la información de una organización está almacenada en forma de documentos digitales (estructurados y no estructurados), esto es una gran cantidad de información que puede ser explorada de manera automática [18]. La MT utiliza técnicas como: categorización de texto, procesamiento de lenguaje natural (PLN), extracción y

recuperación de la información o el aprendizaje automático, seguimiento de tópicos, generación de resúmenes, enlace de conceptos, visualización de información [20].

3.3.8 Desafíos de MT

Como ya hemos estudiado, la minería de textos se enfoca en el análisis de texto no estructurado y la obtención de nuevo conocimiento a partir de un corpus de información. La minería de textos es lograda por las diferentes técnicas que se usan pero no es tan fácil como se pensaría, y algunos desafíos que tiene la minería de textos son los siguientes [20]:

- Número muy alto de posibles dimensiones, todos los tipos de palabras y frases posibles en un lenguaje.
- A diferencia de minería de datos los registros (documentos) no son estructuralmente idénticos. Los registros no son estadísticamente independientes.
- Relaciones complejas y sutiles entre los conceptos de los textos.
- Ambigüedad y sensibilidad al contexto (sinónimos, homónimos, etc.).

3.3.9 Herramientas de MT

La MT puede ser un proceso que se lleve a cabo apoyado de software el cual permite el procesamiento. Entre las herramientas desarrolladas para extraer información, y que intentan inferir relaciones que no aparecen de forma implícita en esa información, pueden citarse:

- ***Natural Language Toolkit*** **(NLTK por sus siglas en inglés):** Es una colección de paquetes y objetos en *python* muy adaptados para tareas de Procesamiento del Lenguaje Natural (PLN). El NLTK es a la vez una herramienta que introduce a nuevas personas al estado del arte en PLN mientras que permite a los expertos sentirse muy cómodos dentro de su entorno. Comparado a otras plataformas, NLTK tiene asunciones por defecto muy fuertes (por ejemplo, un texto es una secuencia de palabras), aunque pueden cambiarse. NLTK no sólo se centra en gente que trabaja en PLN viniendo desde la computación sino también en lingüistas haciendo trabajo de campo.

- ***TextAnalyst*****:** Ayuda rápidamente a resumir, navegar eficientemente, y clasificar los documentos disponibles. Da al usuario la capacidad de realizar una recuperación de datos semántica o de enfocar la exploración del texto hacia un tema determinado. La sinergia de las redes tecnológicas y de la lingüística asegura la velocidad y exactitud en el análisis de textos no estructurados [21].

- **T-LAB:** Es un software de análisis cuantitativo compuesto por un conjunto de herramientas lingüísticas y estadísticas. Está diseñado para plataforma Windows por el psicólogo italiano Franco Lancia y se encuentra en su versión 7.1 [22].

- **PolyAnalist:** Está preparado para trabajar con volúmenes muy grandes de texto y datos estructurados. El software permite realizar categorización, clasificación, predicción, análisis de enlaces, descubrimiento de patrones y detección de anomalías en grandes volúmenes de texto [21].

- **AeroText:** Es una herramienta o biblioteca con aplicaciones de minería de textos para realizar análisis de contenido para textos en diversos idiomas [21].

- **Pimiento:** Esta herramienta en desarrollo está siendo elaborada por la Universidad del País Vasco, no se comercializa, pero se puede obtener una licencia para su uso o su investigación. No se le puede considerar software ya que realmente es una biblioteca o marco para aplicaciones desarrolladas en java para realizar minería de texto. Sus aplicaciones son: c*lustering* de documentos, identificación de idioma en textos y análisis de similitud entre textos [21].

- ***General Architecture for Text Engineering*** **(GATE por sus siglas en inglés):** Es un software de código abierto capaz de resolver casi cualquier problema de procesamiento de texto. *Suite* de herramientas basadas en Java desarrolladas en la Universidad de Sheffield, que comenzó en 1995 y hoy es usada por una amplia comunidad de científicos, compañías, profesores y estudiantes para tareas del procesamiento de lenguaje natural de todo tipo. GATE tiene como objetivo eliminar la necesidad de resolver problemas comunes de ingeniería de texto antes de hacer investigación útil, o reingeniería de procesos antes de convertir los resultados de la investigación en aplicaciones. Las funciones principales de GATE son el modelado y persistencia de estructuras de datos especializadas, visualización y editado de anotaciones, ontologías, árboles de análisis sintáctico, extracción de instancias de entrenamiento para aprendizaje automático, implementaciones de aprendizaje automático.

3.4 Procesamiento del Lenguaje Natural (PLN)

Es esta sección se abordarán temas relacionados con el Procesamiento del Lenguaje Natural (PLN).

3.4.1 Objetivo de PLN

El PLN tiene como objetivo la consecución de sistemas informáticos capaces de implementar funcionalidades que impliquen la comprensión, o la generación de expresiones en lenguaje natural [23]. Podríamos clasificar en tres tipos de objetivos de PLN [24]:

- **Interfaces en lenguaje natural:** ¿No estaría bien dar las órdenes en el mismo lenguaje a todos los ordenadores, y tanto más aún si ese lenguaje fuera uno que los usuarios ya conocieran bien, como su propio lenguaje natural nativo? Esta era la premisa en que se basaban las interfaces en LN hasta finales de los ochenta. No obstante, algunas modernas interfaces gráficas basadas en iconos se están volviendo más fáciles de usar y a veces superan la velocidad de escritura de muchos usuarios. Actualmente, parece que una solución más deseable para cubrir las necesidades de los colectivos de usuarios sería una tecnología mixta consistente en interfaces híbridas de tipo gráfico/LN y voz/LN o voz/LN/gráfico. Los recientes avances en el procesamiento del lenguaje oral, junto con la tecnología PLN están convirtiendo este tipo de interfaces en una realidad práctica.

- **Procesamiento de textos:** Según se ha estimado en congresos de la IFIP (en inglés *International Federation for Information Processing*), hay en todo el mundo más datos almacenados en forma de texto que en cualquier otra forma (como, por ejemplo, bases de datos relacionales o incluso registros de transacciones bancarias). Las ciencias de la información han abordado el problema de la recuperación probabilística, pero han tropezado con las limitaciones que plantea el sistema de palabras clave en cuanto al grado de precisión en el proceso de recuperación.

  Por otra parte, las necesidades de los usuarios van más allá de la recuperación de información e incluyen la extracción de los datos significativos, la elaboración de resúmenes, etc. Las actuales investigaciones en el campo del PLN intentan abordar estos problemas.

- **Traducción automática:** El objetivo original del PLN ha tomado una vez más la delantera en cuanto a resultados científicos recientes, avances tecnológicos y aplicaciones prácticas. Diversos sistemas multilingües eficaces de TA ya están siendo explotados industrialmente y continuarán evolucionando de manera rápida en un futuro inmediato.

3.4.2 Dificultad principal

El lenguaje natural es localmente ambiguo, y la resolución de ambigüedades es necesaria para un procesamiento eficaz [24]. Las ambigüedades se clasifican en cuatro tipos de niveles:

- **A nivel léxico:** Una misma palabra puede tener varios significados, y la selección del apropiado se debe deducir a partir del contexto oracional o conocimiento básico. Algunas investigaciones en el campo del PLN han estudiado métodos de resolver las ambigüedades léxicas mediante diccionarios, gramáticas, bases de conocimiento y correlaciones estadísticas.
- **A nivel referencial:** La resolución de anáforas y catáforas implica determinar la entidad lingüística previa o posterior a que hacen referencia.
- **A nivel estructural:** Se requiere de la semántica para desambiguar la dependencia de los sintagmas preposicionales que conducen a la construcción de distintos árboles sintácticos.
- **A nivel pragmático:** Una oración, a menudo, no significa lo que realmente se está diciendo. Elementos tales como la ironía tienen un papel importante en la interpretación del mensaje.

Para resolver estos tipos de ambigüedades y otros, el problema central en el PLN es la traducción de entradas en lenguaje natural a una representación interna sin ambigüedad**.**

3.4.3 Herramientas de PLN

Existen varias herramientas para poder lograr un PLN (algunas ya mencionadas en la sección 3.3 de MT). Y otras como “COES: Herramientas para Procesamiento de Lenguaje Natural en español” por mencionar alguna. Este trabajo hace énfasis en la utilización de NLTK

ya que en esta herramienta tiene la ventaja que funciona para la extracción de información en texto no estructurado y la implementación de PLN.

3.4.4 Componentes de PLN

El procesamiento del lenguaje natural requiere la realización de las siguientes tareas[24]:

- **Análisis morfológico:** El análisis de las palabras para extraer raíces, rasgos flexivos, unidades léxicas compuestas y otros fenómenos.

- **Análisis sintáctico:** El análisis de la estructura sintáctica de la frase mediante una gramática de la lengua en cuestión.

- **Análisis semántico:** La extracción del significado de la frase, y la resolución de ambigüedades léxicas y estructurales.

- **Análisis pragmático:** El análisis del texto más allá de los límites de la frase, por ejemplo, para determinar los antecedentes referenciales de los pronombres.

- **Planificación de la frase:** Para generar texto, la decisión de cómo estructurar cada frase con el fin de expresar el significado adecuado.

- **Generación de la frase:** La generación de la cadena lineal de palabras a partir de la estructura general de la frase, con sus correspondientes flexiones, concordancias y restantes fenómenos sintácticos y morfológicos.

3.4.5 Aplicaciones de PLN

Las aplicaciones entorno a las que se han desarrollado sistemas de PLN son múltiples; la traducción automática, los interfaces en lenguaje natural a bases de datos y a sistemas expertos, la corrección inteligente de textos, la generación automática de resúmenes, por mencionar algunas [23].

Un factor común de todas las aplicaciones es su potencial de tener alta rentabilidad y deseabilidad. Otra característica común es la dificultad de la implementación de los sistemas de este tipo.

3.4.6 Recursos de PLN

El primer paso para el procesamiento informático del conocimiento lingüístico es la representación formal de dicho conocimiento. Se han creado múltiples recursos para representar la información propia del lenguaje natural, como, por ejemplo:

- **Bases de datos léxicas:** Colección de información lingüística. Cómo las base de datos, se organiza según un determinado modelo que posibilita el almacenamiento, recuperación y modificación de los datos que contiene [25].

- **Corpus:** Un corpus está compuesto por textos producidos en situaciones reales y la inclusión de los que lo componen debe estar guiada por una serie de criterios lingüísticos explícitos para asegurar que pueda usarse como muestra representativa de una lengua [26].

- **Tesauros:** Un tesauro es utilizado al analizar la problemática de traducir los conceptos y sus relaciones, tal como se expresan en los documentos, a un lenguaje con mayor precisión y sin ambigüedades, para facilitar la recuperación de información [30].

- **Ontologías:** Ontologías son recursos construidos que permiten representar el conocimiento compartido y común sobre algo [27].

Estos ejemplos son por mencionar algunos, otros recursos se basan en el uso de herramientas informáticas para realizar determinados análisis focalizados en alguna tarea específica del procesamiento del lenguaje natural, por ejemplo, para el análisis gramatical se encuentran una serie de programas que permiten segmentar un texto, identificar categorías gramaticales (*Part of Speech*, POS por sus siglas en ingles) y establecer las formas raíz de una palabra.

3.5 Clustering

En la siguiente sección se abordan definiciones y conceptos relacionados con *clustering*.

3.5.1 Definición de *clustering*

Existen varias definiciones de c*lustering* entre ellas se encuentran:

- *Clustering* es el proceso de agrupar los datos en clases o en *clusters*, de tal forma que, los datos de un mismo *cluster* tienen una alta similitud y a su vez, son muy diferentes de los de otro *cluster*. Un *cluster* de objetos puede ser tratado colectivamente como un grupo o ser considerado como una forma de compresión de datos [28].

Por otro lado en *Machine Learning* concibe al *clustering* como ~~es~~ un ejemplo de aprendizaje no supervisado a diferencia de la clasificación, el *clustering* o aprendizaje no supervisado no requiere clases predefinidas (ni conjuntos de entrenamiento) [28].

3.5.2 Objetivo de *clustering*

El objetivo de *clustering* es ordenar las observaciones en grupos tales que el grado de asociación natural es alto entre los miembros del mismo grupo y bajo entre miembros de grupos diferentes.

3.5.3 Ventajas de *clustering*

Al realizar *clustering* se puede obtener ventajas significativas, ya que se logra identificar regiones densas y regiones dispersas en el espacio de características, y por lo tanto, descubrir distribución de patrones y correlaciones entre los atributos, al mismo tiempo de que se puede tener la ventaja de detención de anomalías [28].

3.5.4 Algoritmos de *clustering*

El tipo particular de entrada no hace ninguna diferencia para el algoritmo de *clustering*. El algoritmo tratará todas las entradas como un conjunto de números o vectores n-dimensionales. En muchas aplicaciones diferentes, la mejor definición depende del tipo de datos y los resultados deseados.

Algunos algoritmos de *clustering* son [29]:

- **Métodos Jerárquicos:** Algoritmos aglomerativos y algoritmos divisivos.
- **Métodos de Particionado y Recolocación:** C*lustering* probabilístico, métodos de los k-vecinos (k-medoids), métodos de las k-medias, y algoritmos basados en densidad.
- Métodos Basados en Rejillas
- Métodos Basados en la Co-Ocurrencia de Datos Categóricos
- *Clustering* Basado en Restricciones
- **Algoritmos para Datos de Grandes Dimensiones:** C*lustering* subespacial y técnicas de Co-*Clustering*.

3.6 Clasificación de Palabras

Las palabras de un texto libre se pueden clasificar dependiendo de su estructura gramatical como:

- Sustantivos
- Adjetivos
- Verbos
- Pronombres
- Adverbios
- Preposiciones
- Interjección
- Conjunciones.

La MD utiliza un algoritmo llamado *Part of speach* (por sus siglas en ingles POS) que se encarga de la clasificación y etiquetado de cada palabra. La Tabla 1 muestra un ejemplo de la categorización [30]:

(posesivo) adjetivo	sustantivo	verbo	conjunción	verbo	adverbio	preposición	(demostrativo) adjetivo	sustantivo
Mi	Hijo	vive	y	trabaja	aquí	en	esta	ciudad

interjección	Verbo	Adjetivo	conjunción	Pronombre	Verbo	Adjetivo (articulo)	sustantivo
oh!	es	maravilloso	cuando	ella	toca	el	violín

Tabla 1. Ejemplos de clasificación de palabras por categoría.

3.6.1 Etiquetas de la clasificación

Las etiquetas de cada palabra dependen de la categorización que tienen, como se describe al inicio de esta sección, se tiene diferente categorización, a continuación, se presenta las etiquetas que se utilizaran al aplicar el algoritmo POS. Para cada categoría se presentan los atributos, valores y códigos que se toman [31][23]. La Tabla 2 muestra un ejemplo de cómo se representa la etiqueta:

Etiquetas			
Posición	**Atributo**	**Valor**	**Código**
Columna 1	Calumna2	Columna 3	Columna 4

Tabla 2. Ejemplo de cómo se representa la etiqueta

En la columna 1 de la tabla 2, se encuentra un número que hace referencia al orden y posición en que aparecen los atributos. La columna 2 hace referencia a los atributos, el número de los cuales varía dependiendo de la categoría. En la columna 3 se encuentran los valores que puede tomar cada atributo y, finalmente, la columna 4 se representa los códigos que se han establecido para su representación. Las etiquetas en sí sólo son los códigos (columna 4) y se sabe a qué atributo pertenecen por la posición (columna 1) en la que se encuentran.

3.7 N-Gramas

Es un conjunto de palabras que forman una frase. Un n-grama puede ser formado por la combinación de palabras tal como aparecen en un documento. En este caso la letra N indica cuántos elementos debe tomarse en cuenta, es decir, cual es la longitud de la secuencia o del n-grama. Por ejemplo, unigrama (1-grama, en este caso es lo mismo que una palabra), bigrama (2-gramas), trigramas (3-gramas), etc. [24].

En la Tabla 3 se puede ver un ejemplo de lo mencionado anteriormente, se utiliza la frase “Rosa trabaja por las mañanas”, los 2-gramas y 3-gramas, creados son:

2-gramas	3-gramas
Rosa trabaja	Rosa trabaja por
trabaja por	trabaja por las

por las	por las mañanas
las mañanas	

Tabla 3. Ejemplo de N-gramas con N=2, N=3.

Como se puede observar el procedimiento es muy sencillo y se utiliza en el sistema de lingüística computacional.

3.7.1 Frecuencia de N-gramas en un documento

La frecuencia de n-gramas no es tan común en los documentos ya que existe una probabilidad menor de encontrarlos a diferencia de una palabra, esto es esperado ya que se observa la aparición de la misma secuencia de palabras en el documento.

0

3.7.2 Elementos que conforman un N-grama

Para poder clasificar los elementos que constituyen un n-grama se utiliza las clases gramaticales POS, tales como sustantivos, verbos, etc.

3.8 Nube de palabras

Nube de palabras (*Word Cloud*), algunas veces también es llamada con una nube de etiquetas, muestra los miembros de una dimensión elegido como texto, pero en diferentes tamaños y colores, en función de una o dos medidas. Un ejemplo común de uso de la palabra nube está analizando la eficacia de las palabras clave del motor de búsqueda en visita de las métricas de sitio web [32]. En la Figura 8 se muestra un ejemplo de la nube de palabras.

Figura 8. Ejemplo de nube de palabras.

3.9 Bolsa de palabras

Bolsa de palabras (B*ag of Words*) es un método que se utiliza en el procesado del lenguaje para representar documentos ignorando el orden de las palabras. En este modelo, cada documento parece una bolsa que contiene algunas palabras. Por lo tanto, este método permite un modelado de las palabras basado en diccionarios, donde cada bolsa contiene unas cuantas palabras del diccionario. Las principales ventajas de utilizar este modelo es su facilidad de uso y su eficiencia computacional [33].

4. Construcción de la memoria organizacional

Como se ha mencionado en la sección 3.1, una memoria organizacional expresa el conocimiento, describe experiencias adquiridas durante la realización de proyectos. En esta tesis se construye una memoria organizacional tomando la información que se encuentra en los reportes PAP, recordando que la información descrita en dichos reportes muestra el conocimiento reportado por los alumnos acerca de los proyectos que realizaron. Además, la memoria organizacional que se desarrolla, cuenta con técnicas de minería de texto, obteniendo información de los reportes de manera digerible para los usuarios.

4.1. Definición de la memoria organizacional

La definición de la memoria organizacional se basa en la Figura 3 de la sección 3.1, donde el ambiente de decisiones es la coordinación PAP, la organización son los departamentos, las bases de almacenaje son los alumnos. En este caso no tenemos archivos externos, ya que solo se trabaja con los documentos del reporte final. La Figura 9 muestra la estructura de la memoria organizacional de nuestro caso de estudio.

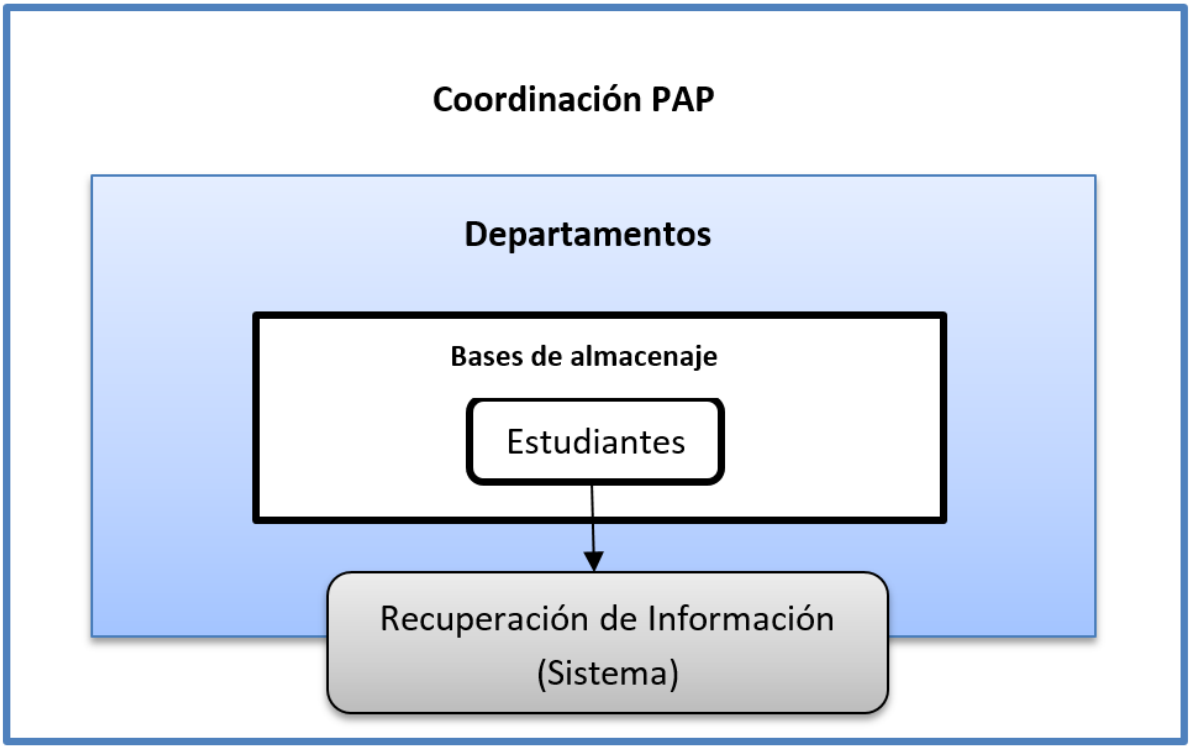


Figura 9. Estructura de la memoria organizacional de este caso de estudio.

4.2. Conocer y obtener información de los proyectos PAP

Como se ha mencionado, la coordinación PAP es la encargada de supervisar y llevar el seguimiento de los proyectos que se crean por los profesores y se realizan por los estudiantes en la asignatura PAP. Por este motivo, como punto de partida se asistió a reuniones con miembros de la coordinación PAP para conocer el proceso que realizan.

La siguiente etapa consistió en recaudar información respecto al tipo de conocimiento que

quería obtener la coordinación. Este punto fue interesante tanto para la coordinación PAP como para mí, ya que se necesita encontrar un balance entre las ventajas y/o desventajas que tienen en su proceso y el beneficio que se quiere obtener con la construcción de la memoria organizacional. Las siguientes actividades son las requeridas por la coordinación PAP:

- Identificar categorías, frases o palabras que nos interesan emplear en el sistema (minería de textos).
- Insumos de documentos corpus: Reportes PAP
- Identificar qué tanto y de qué manera las competencias socio-profesionales, contenidos y temas, son mencionadas por los estudiantes.
- Identificar la atención a problemáticas de la sociedad según la ubicación geográfica
- Aprendizajes mencionados por estudiantes y su relación con otras variables. Para este punto se toma las siguientes frases descritas en la Tabla 4:

Categoría	Sub categorías	Frases o palabras
Aprendizajes	Profesionales Técnicos disciplinares Sociales Personales	me di cuenta, aprendí (aprender), utilice, me ayudo, pude, en lo personal, me da orgullo
Contribución, soluciones, productos, resultados, logros		se desarrolló se realizó se obtuvo se logro se diseñó se propuso se desarrolló se implementó se evaluó se creo se innovo se resolvió

Tabla 4. Aprendizajes mencionados por estudiantes.

4.3. Selección de algoritmos

Los algoritmos que se aplican y que nos ayudan a crear la memoria organizacional son:

- Tokenización
- Frecuencia de palabras
- Generación de n-gramas
- Clasificación de palabras.

Cada uno se ha seleccionado en base a lo que la coordinación PAP desea conoce, descrita en la sección 4.2.

4.4. Análisis de tecnologías

Se han analizado algunas tecnologías como: .Net(C#), PHP, Java, HTML, Python, MySQL, SQL. La Tabla 5 muestra sus ventajas y desventajas:

Nombre	Descripción	Ventajas	Desventajas
.Net (C#)	Microsoft C# es un nuevo lenguaje de programación diseñado para crear un amplio número de aplicaciones empresariales que se ejecutan en .NET Framework.	Es sencillo, moderno, proporciona seguridad de tipos y está orientado a objetos. Se compila como código administrado. Integra el diseño e implementación de formularios de Windows.	Necesitas licencia para desarrollar (Visual Studio).
PHP	PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. PHP se utiliza para generar páginas web dinámicas.	Es un lenguaje multiplataforma. Lenguaje gratuito.	El código fuente no pueda ser ocultado.
HTML	HTML es el lenguaje que se emplea para el desarrollo de páginas de internet. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla	Es el lenguaje de formateo para los navegadores web. Es fácil de entender y utilizar. Su uso es muy extendido.	No tiene semántica. Uso de etiquetas con nombres diferentes. El contenido no puede ser reconocido ni procesado por programas. Tiene un costoso mantenimiento de las páginas. No tiene estándares comunes. Solo tiene hiperenlaces simples
SQL Server	SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial.	Es un sistema de gestión de base de datos. Es útil para manejar y obtener datos de la red de redes. Nos permite olvidarnos de los ficheros que forman la base de datos. SQL permite administrar permisos a todo.	Utiliza mucho la memoria RAM para las instalaciones y utilización de software. No se puede utilizar como practicas porque se prohíben muchas cosas, tiene restricciones en lo particular. Tiene muchos bloqueos a nivel de página, un tamaño de página fijo y demasiado pequeño, una pésima implementación de los tipos de datos variables.
MySQL	Es un sistema de gestión de bases de datos relacional	Es *Open Source*.	Un gran porcentaje de las utilidades de MySQL no están

	desarrollado bajo licencia dual GPL/Licencia comercial por Oracle *Corporation* y está considerada como la base datos *open source* más popular del mundo.	Velocidad al realizar las operaciones. Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema. Facilidad de configuración e instalación. Soporta gran variedad de Sistemas Operativos.	documentadas. No es intuitivo, como otros programas (ACCESS).
Java	Es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.	Manejo automático de la memoria. Lenguaje Multi-plataforma Puede correr en el explorador y en dispositivos móviles. Fácil de aprender. Lenguaje gratuito.	Requiere un intérprete. Algunas implementaciones y librerías pueden tener código rebuscado. Una mala implementación de un programa en java, puede resultar en algo muy lento. Algunas herramientas tienen un costo adicional.
Python	Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, es multiplataforma.	Rápido de desarrollar. Sencillez y velocidad. Sus bibliotecas hacen gran parte del trabajo. Soporta varias bases de datos.	Los programas interpretados son más lentos que los compilados. Declaraciones indentadas (no {}).

Tabla 5. Tecnologías que se pueden usar para crear una memoria organizacional.

Se tomaron varias consideraciones para realizar la implementación de los algoritmos, algunas de ellas son el dominio de los lenguajes de programación tanto para hacer el procesamiento como para realizar la interfaz con el usuario y poder construir la memoria organizacional.

Para poder hacer uso de la memoria organizacional de manera dinámica se ha creado una interfaz para poder manipular el contenido de la memoria y poder extraer la información de manera legible. Dicha interfaz se ha desarrollado en lenguaje .NET (C#) y usa MySQL como motor de base de datos haciendo llamadas a script desarrollados en Python 2.7.

4.5 Implementación

En esta sección se explica cómo son usados e implementados los algoritmos, las modificaciones y adecuaciones que se hacen para lograr una completa integración con el sistema que va a manejar las llamadas a los mismos. La Figura 10 muestra el proceso que se aplica para la construcción de la memoria organización tomando en cuenta desde la introducción de los documentos, hasta la presentación de resultados.

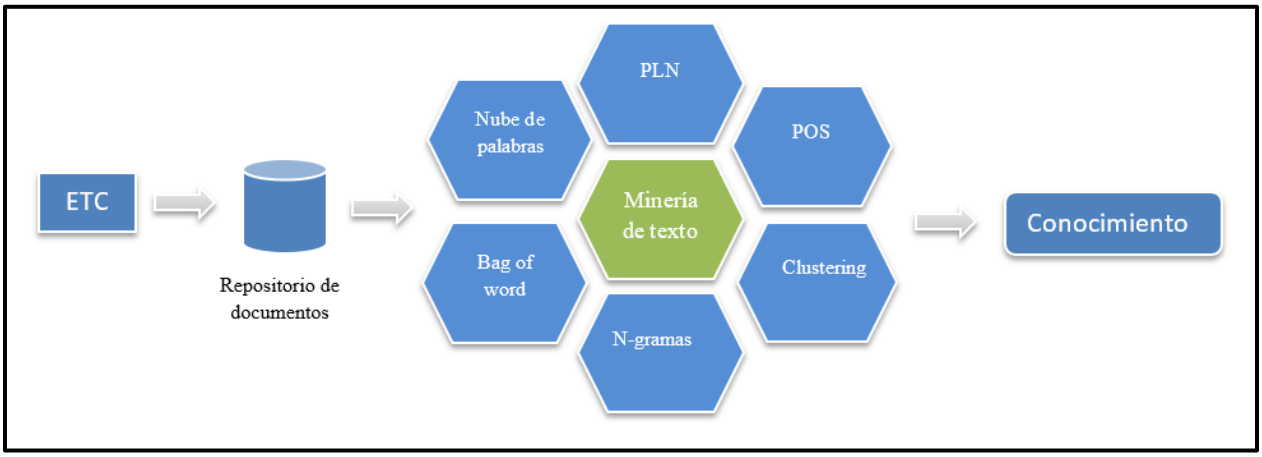


Figura 10. Proceso para crear una memoria organizacional a partir de texto no estructurado usando herramientas de minería de texto.

4.5.1. Extracción, transformación, carga (ETC).

Extracción del texto del documento: para convertir un documento de formato pdf a txt se implementa la biblioteca de python llamada "pdf2txt.py".

```
C:/Python27/Scripts/pdf2txt.py -o "Nombre_documento.txt" -t text "Nombre_Doc_pdf.pdf"
```

Transformación (limpiar texto): una vez que se tiene el texto en formato txt se procede a quitar caracteres especiales, quitar signos de puntuación (excepto la coma y el punto), quitar números y convertir el texto en minúsculas (código en anexo E.2).

Antes:

Los socios menores realizan por lo menos un ahorro de 20 pesos al mes, mientras que los socios mayores ahorran un mínimo de 20 pesos a la semana.

El socio mayor ingresa con 860 pesos, 500 pesos como parte social y 300 pesos de aportación PROFUN, 10 mil pesos que se le otorga al beneficiario en caso de fallecimiento del socio y por último una aportación emergente de 60 pesos para recapitalizar el fondo PROFUN.

Una de las características de la cooperativa es la sencillez, por eso están establecidos en una casa que se acondicionó para utilizarla como oficina.

Como un incentivo para los socios mayores, la cooperativa realiza talleres enfocados al crecimiento personal y sin un cargo extra. Además, los jueves extienden una invitación a todos los socios para sesiones informativas.

Misión

Apoyar la economía de nuestra comunidad fomentando el ahorro y el crédito a través de la unión buscando generar un beneficio colectivo. Nos caracterizamos por ser una cooperativa consciente de las necesidades de nuestros socios y lo suficientemente responsable para atender sus necesidades.

Figura 11. Texto antes de la limpieza.

Después:

instituto tecnologico y de estudios superiores de occidente ac. reporte globalfinal pap en empresas de alta tecnologia proginnt objetivo espe
..... capitulo antecedentes y problematica actual .....
..... enunciado del trabajo .....
.... recursos humanos .....
..... . beneficios esperados y conseguidos del logro de los objetivos ..... conclusiones y
sarrollar un puesto de ingenieria en la industria electronica. el desarrollo electronico en la empresa se divide principalmente por las areas de trabajo
oftware tomando en cuenta los requerimientos establecidos por cada proyecto basado en microcontroladores de bits sensores inercial presion proximidad to
olver problemas. el equipo nacuri se centro inicialmente en el establecimiento de una empresa en un nicho muy especifico. comenzando con el desarrollo de s
esarrollada previamente por una empresa de eua dedicada a ofrecer soluciones de la salud. el principal requisito de dicho diseno de optimizaciones y mejo
onal. trabajos previos demuestran la capacidad de la empresa para llevar a cabo proyectos buscando siempre la referencia de la tecnologia recientemente impl
ing para el diseno de soluciones en electronica con la empresa que ahora es su principal cliente freescale. sin embargo motivado por los buenos resultados a
ente. cuando el pers esta conectado con un monitor de paciente permite la notificacion inmediata de los cuidadores o los servicios de emergencia en respuest
utilicen deben estar deshabilitados uart uart adc spi ssi mcu debe estar en el modo de baja potencia mayor parte del tiempo. solo una interrupcion kbi bo
ro de desarrollo y pruebas de software sera llevado a cabo por desarrolladores de nacuri alan alvarado andoni gonzalez y octavio munoz. diego garay tomara
el camino a seguir para realizar las mejoras necesarias en el nuevo diseno. una vez que se conozcan las mejoras se debera realizar un nuevo diseno para imp
de alta tecnologia proginnt nacuri personal emergency response system v. directora gerente jefe o lider del proyecto octavio munoz solano a. responsabili
cana donde se apoya con el desarrollo de mejoras en la funcionalidad de su producto esperado requiere. esto para nacuri es muy importante pues se espera log
ida de la bateria y acercarla a meses en su funcionamiento regular. proposito o justificacion del proyecto se busca trabajar en conjunto con la empresa cl
hacer uso de las herramientas de diseno libres de freescale para el desarrollo del proyecto. el diseno del producto final no requiere contar con los estanda

Figura 12. Texto después de implementar la limpieza de texto.

4.5.2. Repositorio de documentos

Los documentos son almacenados en formato de texto y son guardados en la base de datos (ver anexos para revisar la estructura de la base de datos). Los documentos son organizados dependiendo su clasificación como son apuesta, programa y periodo (ver anexo x).

4.5.3. Herramientas de minería de texto

Para la implementación de los algoritmos de minería de texto que se utilizan para la creación de la memoria organizacional, es importante contar con una limpieza del documento (sección 4.5.2), partimos de este punto para aplicar los algoritmos.

4.5.3.1. Tokenización (tokenizing)

Se implementó en Python, separando las palabras por el limitador de un espacio en blanco (ver anexo E.3).

4.5.3.2. Crear N-gramas

Para la creación de n-gramas se usa código en Python. Se han creado unigramas,

bigramas, trigramas y cuatrigramas, el código se ha desarrollado en Python (ver anexo E.4).

4.5.3.3. Clasificación de palabras (POS)

Para la clasificación de palabras usamos etiquetas (ver sección 3.6). Se trabaja con el texto limpio para su clasificación y el código se ha desarrollado en Python (ver a anexo E.5).

4.5.3.4. Identificación de competencias

Para la identificación de las competencias se usa código de C#, se encuentra las oraciones que contengan las competencias a revisar (ver anexo E.6).

4.5.3.5. Frecuencias

Para sacar la frecuencia de las palabras y frases se usa código en Python y se almacenan en la base de datos (ver anexo E.7).

4.5.3.6. Nube de palabras

Se crean a partir de la frecuencia (ver anexo E.8).

4.5.4. Conocimiento

En esta sección se elaboran reportes respecto los resultados e información que se obtuvo de la aplicación de los algoritmos. Estos reportes son construidos desde C# y mostrados en la interfaz.

4.5.5. Elaboración del sistema

El sistema se ha basado en las actividades mencionadas en las secciones 4.5.1, 4.5.2, 4.5.3 y 4.5.4. A continuación se presentarán las principales actividades del sistema.

Figura 13. Sistema PAP.

Mapa de actividades

- **Categorías:** Apuesta estratégica, Asignatura, Proyecto PAP, Períodos, Entidad.

- **Búsquedas:**
- **Reportes:** Documentos PAP, Competencias
- **Documentos PAP:** Agregar Documento

Descripción del menú

Categorías: En esta sección se hace un catálogo de las siguientes opciones, de esta manera pueden agregar, modificar dichas categorías:

- Apuesta estratégica
- Asignatura
- Proyecto PAP
- Periodos
- Entidad

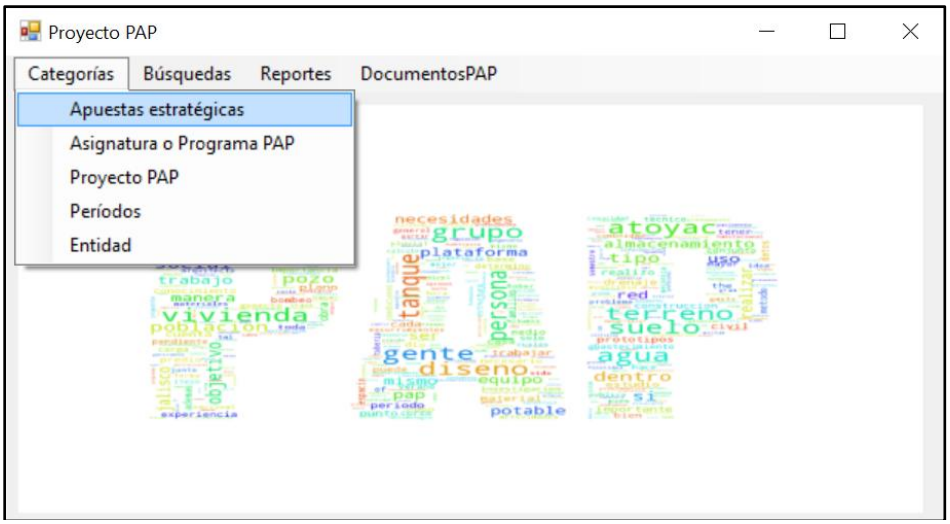


Figura 14. Menú de categorías.

- **Búsquedas:** En esta opción se podrá hacer la búsqueda de palabras y frases (máximo de 4 palabras), así como encontrar su respectiva frecuencia, además de crear la gráfica de frecuencias y la nube de palabra.

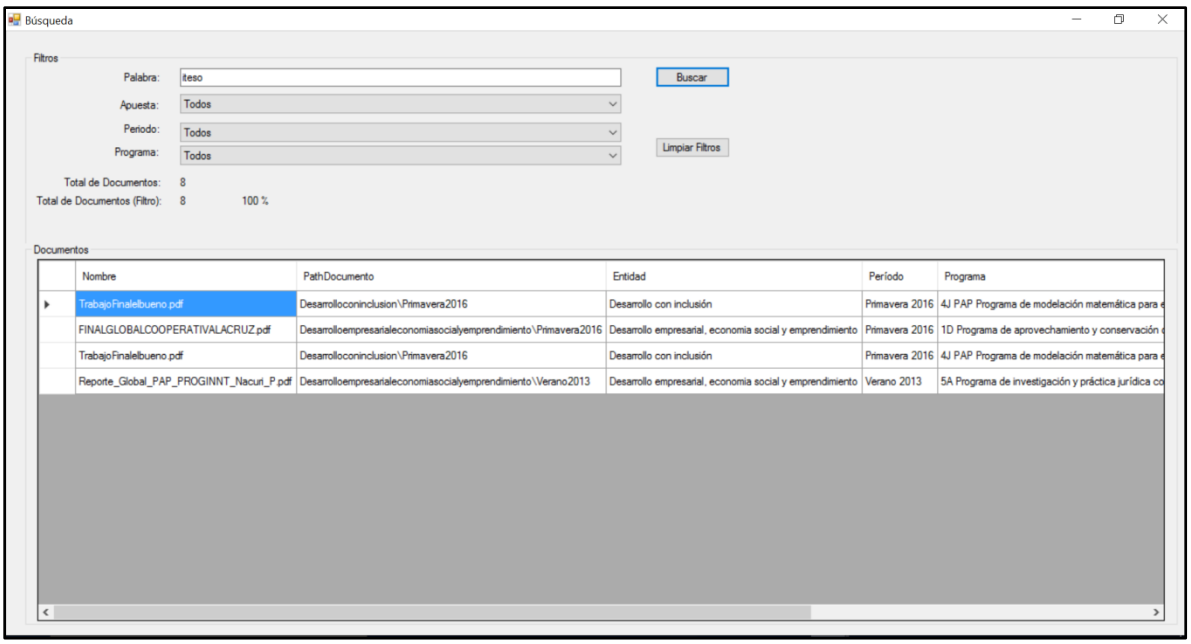


Figura 15. Formulario de búsquedas.

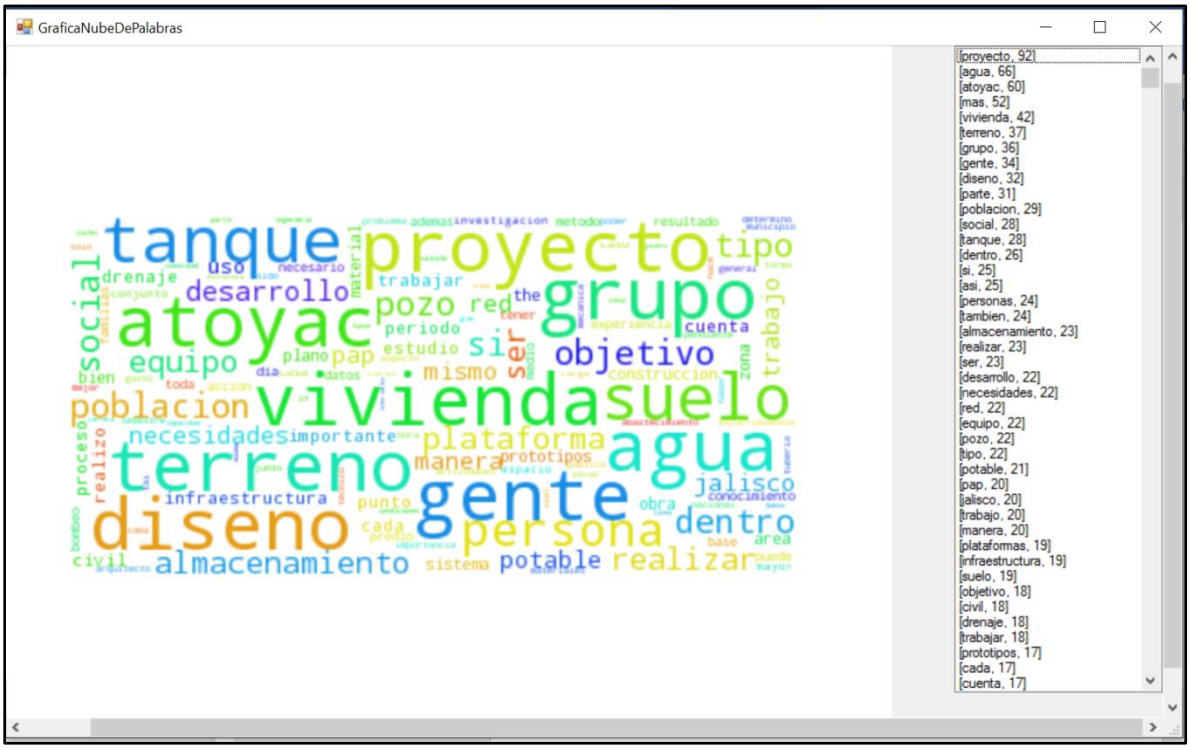


Figura 16. Formulario para mostrar la nube de palabras y su frecuencia.

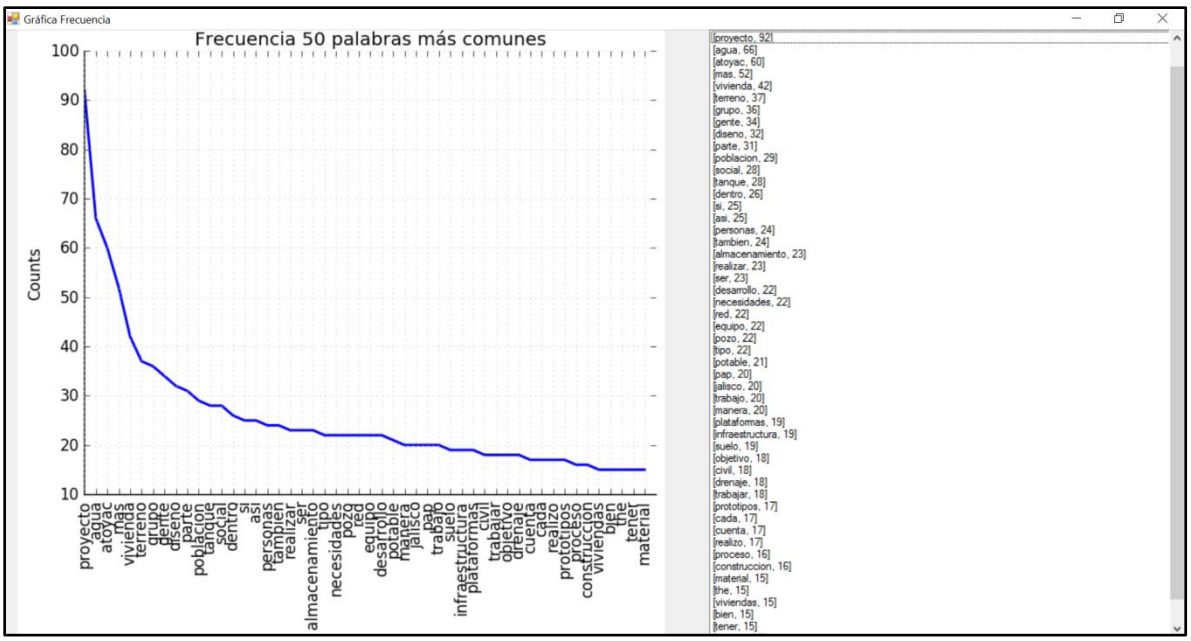


Figura 17. Formulario para mostrar grafica de frecuencia y su frecuencia.

Reportes->Documentos PAP: En esta opción se podrán hacer consultas a los documentos, dependiendo su clasificación por apuesta, período y programas, obteniendo una lista de documentos que cumplan con el filtro seleccionando. Se podrá crear grafica de frecuencia y nube de palabras referente a los documentos que se obtuvieron en la consulta.

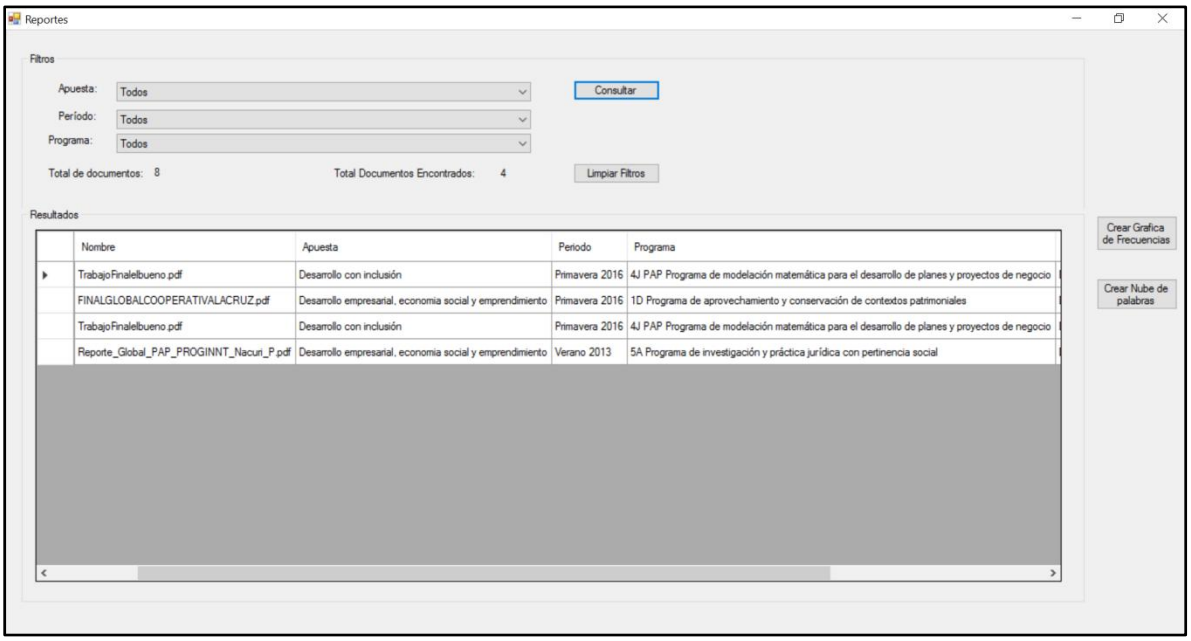


Figura 18. Formulario para la consulta y creación de gráficas.

Reportes ->Competencias: En esta opción podrás obtener un reporte de las competencias que se mencionan en el documento, así como configurar las competencias (agregar, editar).

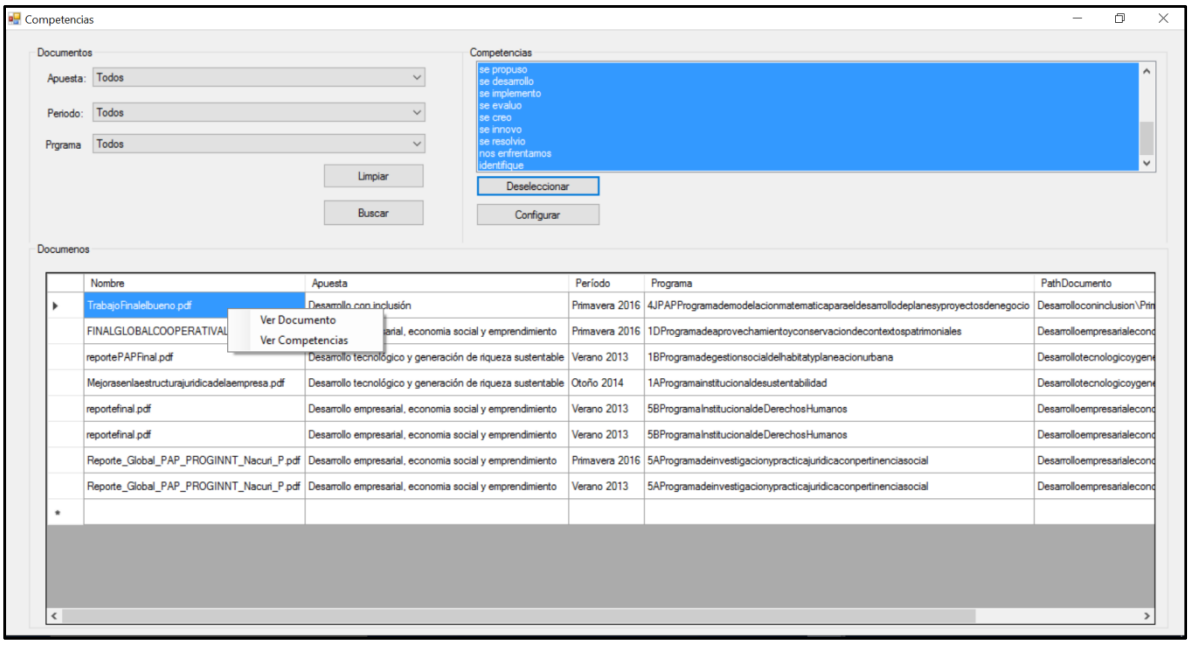


Figura 19. Formulario para consultar las competencias.

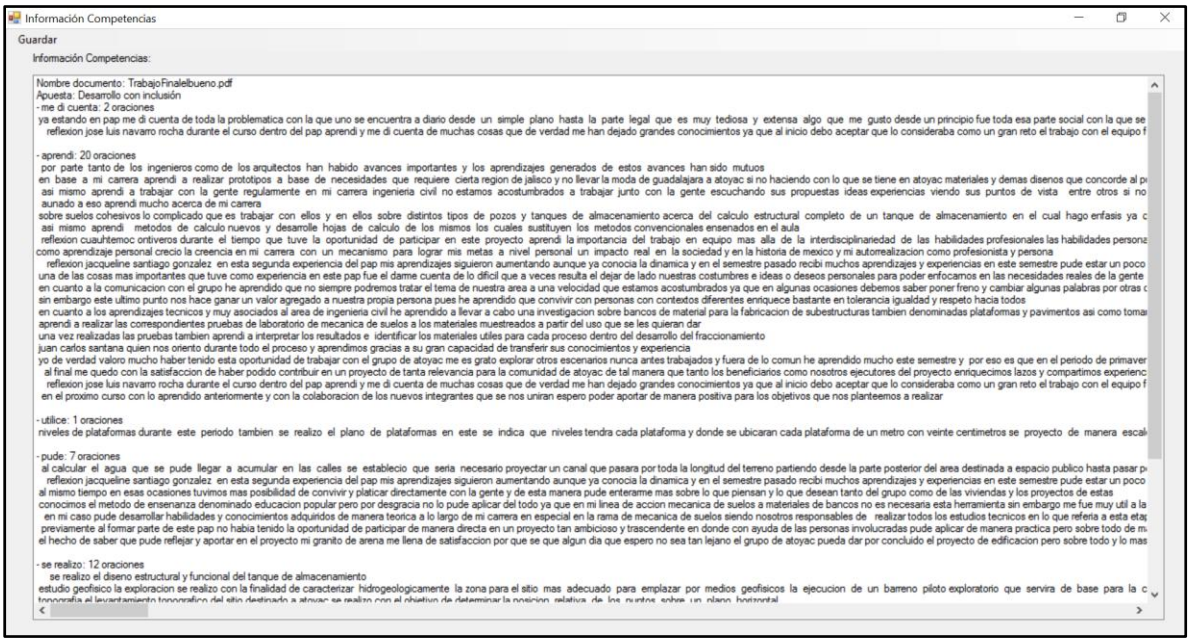


Figura 20. Reporte de competencias encontradas en el documento.

Documentos PAP->Agregar Documento: En esta opción podremos agregar un documento a la memoria organizacional, seleccionando una apuesta, un período, un programa, y una entidad.

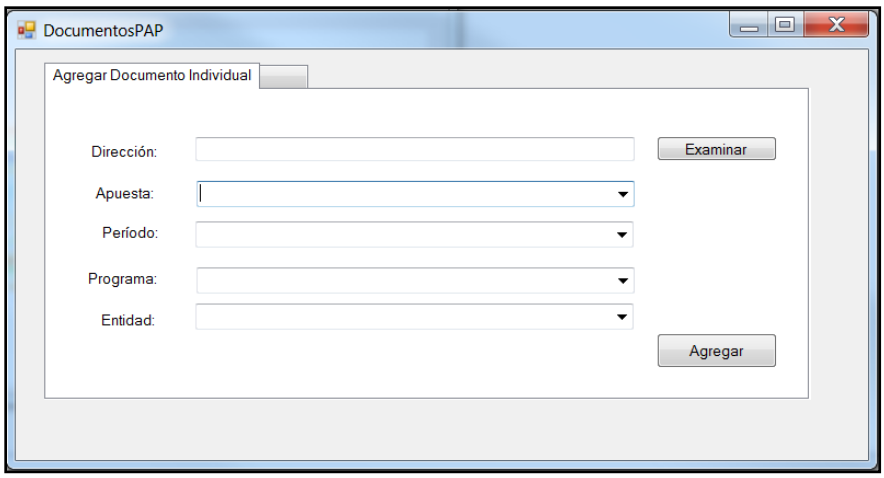


Figura 21. Formulario para agregar un documento a la memoria organizacional.

5. Resultados

5.1 Resultados de la implementación

Los resultados de la implementación son medidos respecto a la sección 1.7 (Hipótesis), donde se obtuvo los siguientes:

- Se han aplicado conceptos y metodologías de minería de texto para extraer información.
- Se han implementados diccionarios que ayudan a la clasificación y descripción de los conceptos para realizar las búsquedas y generar los reportes.
- Se han generado glosario a partir de un repositorio para la clasificación de documentos.
- Se han implementado algoritmos para una clasificación, búsqueda y extracción de información.

5.2 Resultados de la memoria organizacional

Para la CPAP ha sido de gran ayuda el poder extraer información de los PAP desde la herramienta, así como poder consultar los documentos PAP, obtener gráficas, crear reportes y consultar información importante planteada por ellos.

6. Conclusiones

6.1. Conclusiones

A lo largo de la presente investigación y desarrollo del proyecto se ha demostrado como se puede crear una memoria organizacional y extraer información de texto no estructurado. En este caso construir una memoria organización que es alimentado por los documentos PAP y que después se extraiga información usando técnicas y herramientas de minería de texto ha ayudado a la CPAP a la organización de documentos PAP y a la extracción de información que se almacena en dichos documentos, obteniendo información de manera ordenada y clara de temas y competencias específicas.

Por otra parte, vemos que la implementación de las técnicas no es uso exclusivo de un lenguaje en particular ya que si recordamos usamos diferentes lenguajes de programación para implementar los algoritmos.

En general realizar esta investigación y caso de uso práctico para crear una MO usando texto no estructurado aplicando técnicas de minería de texto me ha dejado conocimiento y me muestra un panorama respecto los sectores en el que se pueden aplicar estas técnicas y extraer información que se puede utilizar para toma de decisiones.

6.2. Trabajos futuros

Las actividades en las que estaremos trabajando en un futuro serían las siguientes:

- Análisis de palabra(s) por párrafo.
- Subir información completa del proyecto. (documentos en Excel, PowerPoint, imágenes)
- Filtrar por carrera.
- Conexión con la base de datos de la biblioteca.
- Agregar a los profesores PAP como usuarios.
- Agregar s*top words* desde la interfaz.
- Agregar por carpeta los documentos desde la interfaz.
- Tomar información de la portada para insertar el documento PAP.
- Clasificación de documentos según sus frecuencias.

Bibliografía

[1] G. M. Padilla, “Diseño y operación de los procesos de gestión académica de los reportes de los Proyectos de Aplicación Profesional R PAP,” ITESO, 2012.

[2] L. E. L. HOYOS, “DESARROLLO DE UNA MEMORIA ORGANIZACIONAL PARA GESTIONAR EL CONOCIMIENTO DE LOS PROCESOS CLAVES: CASO KMSOLUCION,” Universidad de Sonora, 2012.

[3] E. Ramos, I. Flores, and H. Nuñez, “Una memoria organizacional para gestionar información y conocimiento de proyectos de investigación de instituciones venezolanas,” vol. 28, pp. 117–131, 2012.

[4] E. L. Riccio and M. C. Gramacho Sakata, “La memoria organizacional para la gestión de conocimiento a través de la utilización de simulación visual,” *JISTEM J. Inf. Syst. Technol. Manag.*, vol. 3, no. 2, pp. 225–255, 2006.

[5] E. W. Stein and V. Zwass, “Actualizing Organizational Memory With Information-Systems,” *Inf. Syst. Res.*, vol. 6, no. 2, pp. 85–117, 1995.

[6] M. de los Á. Martín, “Memoria Organizacional Basada en Ontologías y Casos para un Sistema de Recomendación en Aseguramiento de Calidad María de los Ángeles Martín,” no. 6360, 2010.

[7] R. Reátegui, “Efectividad de los sistemas de memoria organizacional de una institución de educación superior,” *Actual. Investig. en Educ.*, vol. 13, no. 1, pp. 212–239, 2013.

[8] L. C. Molina, “Data mining: torturando a los datos hasta que confiesen,” *Fuoc*, pp. 1–11, 2002.

[9] J. M. Molina and J. García, “Técnicas de análisis de datos,” p. 266, 2006.

[10] K. K. Hirji, “Discovering Data Mining : From Concept to Implementation,” *Acm Sigkdd*, vol. 1, no. 1, pp. 44–45, 1999.

[11] I. . H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. 2005.

[12] Software and hardware solution for professionals, “DATATI,” *2015*. [Online]. Available: http://www.datati.es/herramientas-de-datamining/. [Accessed: 01-Jan-2015].

[13] WEKA the University of Waikato, “WEKA 3: Data Mining Software in Java.” .

[14] R. Hwa, “An Overview of Text Mining What Is Text Mining ? “ The objective of Text Mining is to exploit information,” 2002.

[15] R. Feldman and I. Dagan, “Knowledge Discovery in Textual Databases (KDT).,” *Int. Conf. Knowl. Discov. Data Min.*, pp. 112–117, 1995.

[16] A.-H. Tan, “Text Mining: The state of the art and the challenges,” *Proc. PAKDD 1999 Work. Knowl. Disocovery from Adv. Databases*, vol. 8, pp. 65–70, 1999.

[17] Universidad Carlos III, “Mineria de Textos,” *Mineria de Textos*. [Online]. Available: http://textmining.es/metodolog%C3%ADa de la miner%C3%ADa de textos.html.

[Accessed: 01-Jan-2015].

[18] S. Equihua, “Data & Text Mining,” 2014. [Online]. Available: http://www.infotecarios.com/data-text-mining/. [Accessed: 01-Jan-2015].

[19] L. Villase and M. Montes, “Descubrimiento Automático de Hipónimos a partir de Texto no Estructurado Rosa María Ortega Mendoza Maestra en Ciencias en la Especialidad de Ciencias Computacionales,” 2007.

[20] M. Grobelnik and D. Mladenic, “Text Mining,” *Comput. Linguist.*, pp. 1–112, 2007.

[21] Megaputer, “Making sense of unstructured data,” 2007. [Online]. Available: http://megaputer.com/site/text_mining.php. [Accessed: 01-Jan-2015].

[22] Alberto de Francisco, “Aplicación y viabilidad de uso del software de Análisis Cuantitativo de Textos TLAB 7.1 en el análisis de las representaciones sociales presentes en la web soyborderline.com,” 2010. [Online]. Available: http://pendientedemigracion.ucm.es/info/mediars/MediacioneS6/Indice/Stefanelloyotros2010/stefanelloyotros2010.html. [Accessed: 01-Jan-2015].

[23] M. de B. Rodríguez, *Integración de técnicas de procesamiento del lenguaje natural para la recuperación de información en bibliotecas de componentes software*. 2001.

[24] Jaime Carbonell, “El procesamiento del lenguaje natural, tecnología en transición,” *Congreso de Sevilla*, 1992. [Online]. Available: http://cvc.cervantes.es/obref/congresos/sevilla/tecnologias/ponenc_carbonell.htm. [Accessed: 01-Jan-2015].

[25] Silvia Arano, “La ontología: una zona de interacción entre la Lingüística y la Documentación,” *2*, 2003. [Online]. Available: https://www.upf.edu/hipertextnet/numero-2/ontologia.html. [Accessed: 01-Jan-2015].

[26] Francis de la Caridad Fernández Reyes, *Integración de métodos para la desambiguación del sentido de las palabras*. 2012.

[27] S. Arano, “Los tesauros y las ontologías en la Biblioteconomía y la Documentación,” *Anu. Hipertext.net*, no. 3, pp. 1–14, 2005.

[28] M. C. Montes, “clustering: Clasificación no Supervisada Gráficas estadística y minería de datos con python,” 2013.

[29] G. Sidorov, “N-gramas sintácticos no-continuos,” *Polibits*, no. 48, pp. 69–78, 2013.

[30] Grigori Sidorov, “CONSTRUCCIÓN NO LINEAL DE N-GRAMAS EN LA LINGÜÍSTICA COMPUTACIONAL,” 2013.

[31] “INTRODUCCIÓN A LAS ETIQUETAS EAGLES.” [Online]. Available: http://www.cs.upc.edu/~nlp/tools/parole-sp.html.

[32] G. Peck, *Tableau 8: The Official Guide*. McGraw-Hill, 2013.

[33] C.-F. Tsai, “Bag-of-Words Representation in Image Annotation: A Review,” *ISRN Artif. Intell.*, vol. 2012, pp. 1–19, 2012.

Anexos

A. Resumen de palabras clave

Anáfora: Relación existente entre un elemento del discurso y otro elemento del mismo discurso que ha aparecido formulado anteriormente

Catáforas: Palabra o palabras que sirven para anticipar parte del discurso (texto) que aún no se ha enunciado o mencionado.

Clustering: Clasificación.

Concepto: Los conceptos son las características generadas a partir de una colección de documentos mediante métodos manuales, estadísticos, basados en reglas o híbridos. En comparación con los términos, los conceptos están en un nivel más alto de abstracción.

Corpus: El corpus (en plural corpora) es como se conoce al conjunto de textos estructurados ya preparados para aplicar técnicas de extracción de conocimiento.

Datos estructurados: Datos con formato predeterminado que habitualmente son almacenados en registros con valores sencillos (categóricos, ordinales o continuos) dentro de una base de datos.

Datos no estructurados: Datos sin formato que habitualmente son almacenados en forma de documentos de texto.

Frecuencia de una palabra: El número de veces que una palabra aparece en un documento.

***Hardware*:** Componentes físicos que constituyen la estructura física.

Matriz de ocurrencias: Representación de la relación basada en la frecuencia de palabras de los términos en un documento. Tiene formato de tabla, donde las filas son los términos y las columnas los documentos.

Morfología: La rama de la lingüística y del procesamiento del lenguaje natural encarga del estudio de la estructura interna de las palabras.

Raiz (***Stemming***): Proceso de reducción de los términos a su raíz.

Software: Programas informáticos que hacen posible la realización de tareas específicas dentro de un computador o máquina.

Stop words: También conocidas como *noise words* son aquellas palabras que no aportan información y que se descartan en algún momento del procesamiento del lenguaje natural. No existe una lista oficial de stop *words*, pero la mayoría de herramientas de procesamiento descarta los artículos, verbos auxiliares y palabras dependientes del contexto.

Término: Un término puede ser una única palabra o una frase extraída del corpus mediante técnicas de procesamiento del lenguaje natural (PLN).

Tesauros: Lista de palabras o términos controlados empleados para representar conceptos.

Token: Bloque de texto que se caracteriza por la función que desempeña dentro de una oración.

Tokenizing: El proceso por el cual se divide en *tokens* la oración es lo que se conoce como *tokenizing*.

B. Diagrama de estructura de la base de datos

C. Requerimientos del sistema

Elaborar un sistema para lo coordinación PAP es requerido ya que ayudara a obtener conocimiento que se encuentra en los reportes PAP para toma de decisiones y futuras implementaciones al programa PAP. Los módulos y actividades que podrán usar los usuarios se describen a continuación.

Módulos

- **Presentación**: Contendrá una descripción de lo que la coordinación PAP, así como un hipervínculo al programa PAP.
- **Configuraciones**: En este módulo podrán hacer altas, bajas, modificaciones y eliminaciones a los diferentes catálogos con los que cuenta el sistema.
- **Documentos**: En este módulo podrán dar de alta los documentos.
- **Reportes**: En este módulo podrán ver los reportes. (Grafica de frecuencias)
- **Búsquedas**: En este módulo podrán hacer búsquedas en los documentos dependiendo el filtro que se seleccione.

Descripción de los módulos

Presentación: La presentación traerá una descripción de lo que se hace en la coordinación PAP, a así como un hipervínculo al programa PAP.

Actividades:

- Descargar el programa actual de PAP.

Configuraciones: En este módulo podrán hacer altas, bajas, modificaciones y eliminaciones a los diferentes catálogos con los que cuenta el sistema.

Actividades:

- Dar de altas, bajas, modificaciones referentes a las apuestas.
- Dar de altas, bajas, modificaciones y eliminaciones referentes a los colores de las apuestas.
- Dar de altas, bajas, modificaciones y eliminaciones referentes a los periodos.
- Dar de altas, bajas, modificaciones y eliminaciones referentes a los departamentos.
- Dar de altas, bajas, modificaciones y eliminaciones referentes a los programas. (hacer una gráfica UML de cardinalidad)
- Dar de altas, bajas, modificaciones y eliminaciones referentes a los proyectos.

Documentos: En este módulo podrán alimentar al sistema de documentos a analizar.

Actividades:

- Consultar los documentos que se tienen, el usuario podrá verlos por la siguiente clasificación:
    1. Apuesta.
    2. Departamento.
    3. Programa.
    4. Periodo.

5. Proyecto.

- Consultar y/o generar un reporte dependiendo el filtro.

  NOTA: Revisar manera de mostrar la información., Investigar ->imprimir listado de documentos dependiendo el filtro.
- Dar de altas documentos uno por uno con las siguientes clasificaciones:
  1. Apuesta ->Periodo (experimentos)
  2. Departamento->Programa -> Periodo ->Proyecto

Reportes: En este módulo podrán ver los reportes. (Gráfica de frecuencias)

Actividades:

- Podrán seleccionar el filtro que deseen:
  1. Apuesta ->Periodo (experimentos)
  2. Entidad->Programa -> Periodo ->Proyecto
     Entidad: departamento

Búsquedas: En este módulo podrán hacer búsquedas de una o varias palabras, frases, en los documentos dependiendo el filtro que se seleccione.

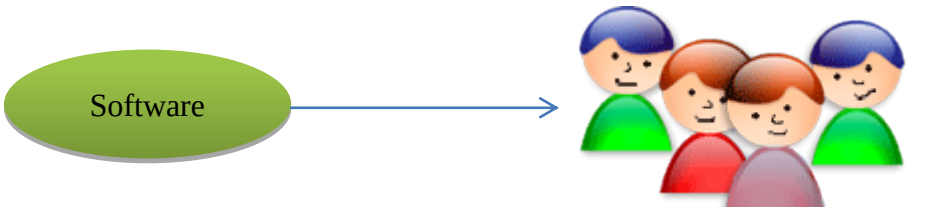
Actividades:

- Podrán seleccionar el filtro que deseen:
  3. Apuesta ->Periodo (experimentos)
  4. Departamento->Programa -> Periodo ->Proyecto

Arquitectura del Sistema

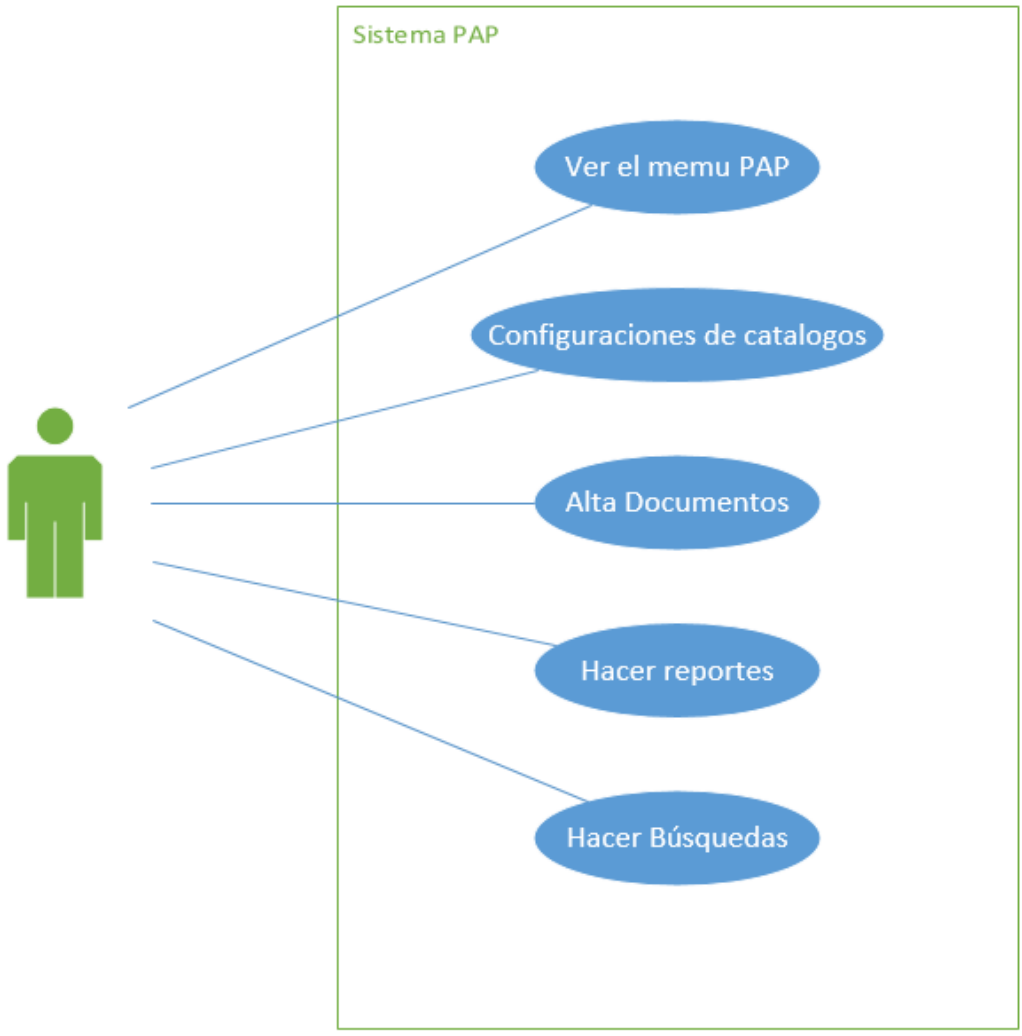
La arquitectura del sistema tenemos dos propuestas:

1. Solo usuarios de la coordinación PAP.



Usuarios coordinación PAP

D. Diagrama de casos de uso del sistema.



E. Código Fuente

E.1. Biblioteca para convertir de formato pdf a txt

```
C:/Python27/Scripts/pdf2txt.py -o “Nombre_documento.txt” -t text “Nombre_Doc_pdf.pdf”
```

E.2. Instrucciones para limpiar el texto

```
public static string QuitarAcentos(string cadena){
        string stFormD = cadena.Normalize(NormalizationForm.FormD);
        int len = stFormD.Length;
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < len; i++){
           System.Globalization.UnicodeCategory uc =
System.Globalization.CharUnicodeInfo.GetUnicodeCategory(stFormD[i]);
           if (uc != System.Globalization.UnicodeCategory.NonSpacingMark)
           {
              sb.Append(stFormD[i]);
           }
        }
        return (sb.ToString().Normalize(NormalizationForm.FormC));
     }

public static string RemoverCaracteresEspeciales(string str){
        StringBuilder sb = new StringBuilder();
        foreach (char c in str){
           if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || c == '.' || c == '_' || c == ' ' || c ==
'ñ'){
              if (c == 'ñ') {
                 sb.Append('n');  }
              else {
                 sb.Append(c); }
           }
        }
        return sb.ToString();
     }
```

E.3. Tokenización

```
tokenize= RegexpTokenizer("[\w']+")
token = tokenize.tokenize(texto)
```

E.4. Crear N-Gramas

```
dosGramass = nltk.bigrams(token)
triGramas = nltk.trigrams(token)
cuatriGramas = ngrams(token,4)
```

E.5. Clasificación de palabras

```
#-*- coding: utf8 -*-
```

```
# about the tagger: http://nlp.stanford.edu/software/tagger.shtml
# about the tagset: nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html
import os
java_path = "C:/Program Files/Java/jdk1.8.0_31/bin/java.exe"
os.environ['JAVAHOME'] = java_path

listaSustantivos = []
listaAdjetivos = []
listaAdverbios = []
listaDeterminantes = []
listaVerbos = []
listaPronombres = []
listaConjunciones = []
listaInterjeccion = []

listaSustantivosUnicos = []
listaAdjetivosUnicos = []
listaAdverbiosUnicos = []
listaDeterminantesUnicos = []
listaVerbosUnicos = []
listaPronombresUnicos = []
listaConjuncionesUnicos = []

def isSustantivo(word,tagS):
        if 'n' in tagS[0]:
                listaSustantivos.append (word)
                listaSustantivosUnicos = set(listaSustantivos)

def isAdjetivo(word,tagA):
        if 'a' in tagA[0]:
                listaAdjetivos.append(word)

def isAdverbios(word,tagA):
        if 'r' in tagA[0]:
                listaAdverbios.append(word)

def isDeterminantes(word,tagD):
        if 'd' in tagD[0]:
                listaDeterminantes.append(word)

def isVerbo(word,tagV):
        if 'v' in tagV[0]:
                listaVerbos.append(word)

def isPronombre(word,tagP):
        if 'p' in tagP[0]:
                listaPronombres.append(word)

def isConjuncion(word,tagC):
        if 'c' in tagC[0]:
                listaConjunciones.append(word)
```

```
#tokenizacion ----------------------------
import sys
from nltk.tokenize import RegexpTokenizer

tokenize= RegexpTokenizer("[\w']+")
texto= ''

infile =
open('C:\wamp\www\phptest\Documentos\Empleabilidadyemprendimiento\Otono2014\Reporte
_Global_PAP_PROGINNT_Nacuri_P2012.txt', 'r')
texto = infile.read()

token = tokenize.tokenize(texto)
#----------------------------------------

import nltk

from nltk.tag.stanford import POSTagger

infile =
open('C:\wamp\www\phptest\Documentos\Empleabilidadyemprendimiento\Primavera2016\Rep
orte_Global_PAP_PROGINNT_Nacuri_P2012 - Copy.txt', 'r')
texto = infile.read()

spanish_postagger = POSTagger('models/spanish.tagger', 'stanford-postagger.jar',
encoding='utf8')

sentences = ['El Zapopan se usa principalmente para sahumar en distintas ocasiones como lo son
las fiestas religiosas bonitas.']
sentences = [texto]
print (sentences)

for sent in sentences:
   words = sent.split()
   tagged_words = spanish_postagger.tag(words)
   nouns = []

   for (word, tag) in tagged_words[0]:
      print(word+' '+tag).encode('utf8')
      #if isNoun(tag): nouns.append(word)
      isSustantivo(word, tag)
      isAdjetivo(word, tag)
      isAdverbios(word, tag)
      isDeterminantes(word, tag)
      isVerbo(word, tag)
      isPronombre(word, tag)
      isConjuncion(word, tag)

listaSustantivosUnicos = set(listaSustantivos)
```

```
listaAdjetivosUnicos = set(listaAdjetivos)
listaAdverbiosUnicos = set(listaAdverbios)
listaDeterminantesUnicos = set(listaDeterminantes)
listaVerbosUnicos = set(listaVerbos)
listaPronombresUnicos = set(listaPronombres)
listaConjuncionesUnicos = set(listaConjunciones)

print(listaSustantivosUnicos)
print(listaAdjetivosUnicos)
print(listaAdverbiosUnicos)
print(listaDeterminantesUnicos)
print(listaVerbosUnicos)
print(listaPronombresUnicos)
print(listaConjuncionesUnicos))
```

E.6. Identificación de competencias

```
private void ObtenerCompetecias(List<string> listaCompetencias, string nombreDocumento,
string apuesta, string textoLimpio)       {
        string[] texto = textoLimpio.Split('.');
        List<string> listaOraciones = new List<string>();
        string resultadosCompetencias = @"Nombre documento: " + nombreDocumento +
Environment.NewLine +
                                @"Apuesta: " + apuesta + Environment.NewLine;

        foreach(string competencia  in listaCompetencias)          {
           string[] competenciasOraciones = Array.FindAll(texto,element =>
element.Contains(competencia));
           if (competenciasOraciones.Length > 0)              {
              resultadosCompetencias = resultadosCompetencias +
AgregarAListaResultados(competencia, competenciasOraciones) + Environment.NewLine;
           }
        }

        InformacionCompetencias informacionCompetencias = new
InformacionCompetencias(resultadosCompetencias);
```

E.7. Frecuencias

```
import sys

import nltk

from nltk import FreqDist
from nltk.tokenize import RegexpTokenizer
from nltk.util import ngrams
from nltk.corpus import stopwords
import MySQLdb
import operator
```

```
id_doc = sys.argv[1]

#configuracion BD
db = MySQLdb.connect(host="localhost", # your host, usually localhost
                     user="root", # your username
                     passwd="xxx", # your password
                     db="proyectospap") # name of the data base

#Consulta los textos
cur = db.cursor()
myQuery = "select d.id_doc, dl.id_documentos_limpios,  dl.texto_limpio from (select id_doc,
id_documento_limpio from documentos where id_doc = " + str(id_doc) + ") d left join (select
id_documentos_limpios, texto_limpio from documentoslimpios) dl on d.id_documento_limpio =
dl.id_documentos_limpios"
print myQuery
cur.execute(myQuery)

# obtener el texto limpio
for row in cur.fetchall() :
    id_doc = str(row[0])
    id_doc_limpio = str(row[1])
    texto = row[2]
    print "id Doc: "  + id_doc
    print "id Doc Limpio: " + id_doc_limpio
    print "texto: " + texto
#-----------------------

#-Se hace la tokenizacion del texto
tokenize= RegexpTokenizer("[\w']+")
token = tokenize.tokenize(texto)
#-----------------------------------

#Se quitan las stopwords que son genericas
stops = set(stopwords.words('spanish'))
tokenStopWords = []
for word in token:
            if  word not in stops:
                        tokenStopWords.append(word)
#print  tokenStopWords
tokenFrecuenciaF = FreqDist(tokenStopWords)

#quitar palabras repetidas
seen = set()
tokenSinRepeticiones = []
for item in tokenStopWords:
    if item not in seen:
        seen.add(item)
        tokenSinRepeticiones.append(item)

#Guardar las frecuencias
```

```
dicTokenFrecuencia = {}
tokenFrecuencia = []
tokenFrecuenciaGuardar = ""
tupla = ""
tokenFrecuenciaStr = ""
for palabra in tokenSinRepeticiones:
    dicTokenFrecuencia[palabra] = str(tokenFrecuenciaF[palabra])
    tupla = palabra + ":" + str(tokenFrecuenciaF[palabra])
    tokenFrecuencia.append(tupla)
    tokenFrecuenciaStr = tokenFrecuenciaStr + tupla +  ","
#-----------------------------------------

#Creacion de los dos-gramas
n = 2
dosGramass = nltk.bigrams(token)
dosGramasFrecuencia = []
dosGramasFrecuenciaStr = ""
fdist = nltk.FreqDist(dosGramass)
for k,v in fdist.items():
    tupla = k[0] + " " +  k[1]
    value = str(v)
    dosGramasFrecuencia.append(tupla +":" + value)
    dosGramasFrecuenciaStr = dosGramasFrecuenciaStr + tupla +":" + value + ","
#---------------------------------------


#-- CReaciond e tri gramas----------------------------
triGramas = nltk.trigrams(token)
tresGramasFrecuencia = []
frecuenciaTriGRamas = nltk.FreqDist(triGramas)
tresGramasFrecuenciaStr = ""
for k,v in frecuenciaTriGRamas.items():
    tupla = k[0] + " " +  k[1] + " " + k[2]
    value = str(v)
    tresGramasFrecuencia.append(tupla +":" + value)
    tresGramasFrecuenciaStr = tresGramasFrecuenciaStr + tupla +":" + value + ","
#-------------------------------------------------

#-- CReaciond e tri gramas----------------------------
listCuatrigramas = []
cuatriGramas = ngrams(token,4)
print cuatriGramas.next()
listCuatrigramas = list(cuatriGramas)
cuatriGramasFrecuencia = []
cuatriGramasFrecuenciaStr = ""
frecuenciaCuatriGRamas = nltk.FreqDist(listCuatrigramas)
for k,v in frecuenciaCuatriGRamas.items():
    tupla = k[0] + " " +  k[1] + " " + k[2] + " " + k[3]
    value = str(v)
    cuatriGramasFrecuencia.append(tupla +":" + value)
    cuatriGramasFrecuenciaStr = cuatriGramasFrecuenciaStr + tupla +":" + value + ","
```

```
#--------------------------------------------------


#Insertar los resultados de la frecuencia en la base de datos
#print "Frecuencia de token: " + tokenFrecuenciaStr
#print "Frecuencia de dos_gramas: " + dosGramasFrecuenciaStr
#print "Frecuencia de tres_gramas: " + tresGramasFrecuenciaStr
#print "Frecuencia de cuatri_gramas: " + cuatriGramasFrecuenciaStr

cur = db.cursor()
myQuery = "Insert into frecuencia(id_doc, id_documento_limpio, token, dos_grama, tri_grama,
cuatri_grama) values (%s,%s,%s,%s,%s,%s)"
print cur.execute(myQuery,(id_doc, id_doc_limpio, tokenFrecuenciaStr,
dosGramasFrecuenciaStr, tresGramasFrecuenciaStr, cuatriGramasFrecuenciaStr))
db.commit()
db.close()

# obtener el texto limpio
for row in cur.fetchall() :
    print str(row[0])
```

E.8. Nube de palabras

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
#!/usr/bin/env python2

import sys
import nltk
import numpy as np
import matplotlib.pyplot as plt
import MySQLdb
import operator
from nltk import FreqDist
from nltk.tokenize import RegexpTokenizer
from nltk.util import ngrams
from nltk.corpus import stopwords
from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

plt.ion()

#sys.argv[0] = 7
id_doc = sys.argv[1]
#id_doc = 18
print ("Doc: " + str(id_doc))

pathServerDocumento = "C:\\Users\\xxx\\";
```

```
#configuracion BD
db = MySQLdb.connect(host="localhost", # your host, usually localhost
                     user="root", # your username
                     passwd="xxxx", # your password
                     db="proyectospap") # name of the data base

#Consulta los textos
cur = db.cursor()
myQuery = "select d.nombre, d.path_documento, dl.texto_limpio from  documentos d  left
join(select id_documentos_limpios, texto_limpio from documentoslimpios) dl on
d.id_documento_limpio = dl.id_documentos_limpios where id_doc = " + str(id_doc)
print myQuery
cur.execute(myQuery)
print cur.execute(myQuery)

# obtener el texto limpio
for row in cur.fetchall() :
    nombreDoc = str(row[0])
    pathDocumento = str(row[1])
    texto = str(row[2])
    #print "nombre: " + nombreDoc
    #print "texto: " + texto
#-----------------------

d = path.dirname(__file__)

coloring = np.array(Image.open(path.join(d, "iteso1.png")))

#print  tokenStopWords
stops= []
stopwords =stopwords.words('spanish')
stops = set(stopwords)
stops = stops | set(['mas','parte', 'tambien', 'asi'])


wc = WordCloud(background_color="white", max_words=2000,
               stopwords =stops,
               max_font_size=40, random_state=42)

# generate word cloud
wc.generate(texto)


# create coloring from image
image_colors = ImageColorGenerator(coloring)

# show
plt.imshow(wc)
plt.axis("off")
```

```
plt.savefig(pathServerDocumento + pathDocumento +"\\" + str(nombreDoc) + "_nube.png")
plt.figure()
```