**Universidad Tecnológica de la Habana**

**“José Antonio Echeverría”**

**Facultad de Ingeniería Informática**



Automatización de la gestión de tareas en documentos institucionales

*Informe de la práctica laboral 2*

|  |  |
| --- | --- |
| ‍‍**Autora:** | Vanesa Casañola Valdés |
| ‍‍ |  |
| **Tutor:** | MSc. Wenny Hojas Mazo |
| ‍ | Profesor asistente de la Facultad de Ingeniería Informática Cujae, La Habana, Cuba |

**La Habana,**

**Agosto, 2024**

Resumen

En las organizaciones modernas, los intercambios internos son fundamentales para la resolución de problemas y la toma de decisiones, pero a menudo son confidenciales y no accesibles al público. Estos intercambios, que abarcan desde la planificación estratégica hasta la resolución de conflictos, se documentan a través de informes detallados. Aunque estos informes facilitan la comunicación interna y la transparencia, la extracción manual de tareas puede ser laboriosa y propensa a errores, especialmente cuando la información es ambigua o sensible. El seguimiento automático de tareas es crucial para evitar la pérdida de información importante y garantizar que se cumplan a tiempo. La automatización mediante algoritmos de aprendizaje automático se presenta como una solución efectiva para mejorar la precisión y eficiencia en la identificación de tareas. En este proyecto se ha desarrollado un sistema que utiliza aprendizaje automático y procesamiento del lenguaje natural para identificar y seguir tareas en documentos de texto de manera automática. El sistema incluye un módulo implementado en Python utilizando spaCy, y las pruebas realizadas han confirmado que el módulo es eficaz en la identificación de muchas tareas, aunque también ha revelado limitaciones, como la dificultad para reconocer tareas con sujetos omitidos y la detección de objetos en ciertas oraciones. El módulo genera archivos con las tareas identificadas, facilitando su gestión. A pesar de algunos errores, el sistema representa un avance al simplificar el tedioso proceso de extracción manual de tareas y reducir la posibilidad de errores, ofreciendo una herramienta útil para mejorar la eficiencia operativa.

Palabras claves: extracción de tareas, seguimiento automático de tareas, aprendizaje automático, procesamiento del lenguaje natural.

Abstract

In modern organizations, internal exchanges are crucial for problem-solving and decision-making but are often confidential and not accessible to the public. These exchanges, which range from strategic planning to conflict resolution, are documented through detailed reports. Although these reports facilitate internal communication and transparency, manually extracting tasks can be laborious and prone to errors, especially when the information is ambiguous or sensitive. Automatic task tracking is essential to prevent the loss of important information and ensure timely completion. Automation through machine learning algorithms emerges as an effective solution to improve precision and efficiency in task identification. This project has developed a system that uses machine learning and natural language processing to automatically identify and track tasks in text documents. The system includes a module implemented in Python using spaCy, and tests have confirmed that the module is effective in identifying many tasks, although it has also revealed limitations, such as difficulty recognizing tasks with omitted subjects and detecting objects in certain sentences. The module generates files with the identified tasks, facilitating their management. Despite some errors, the system represents a significant advance by simplifying the tedious process of manual task extraction and reducing the possibility of mistakes, providing a useful tool to enhance operational efficiency.

Keywords: task extraction, automatic task tracking, machine learning, natural language processing.

Índice

[Introducción 1](#_Toc176742737)

[Capítulo 1: Fundamentos teóricos 6](#_Toc176742738)

[1.1 Documentación de intercambio internos en instituciones 6](#_Toc176742739)

[1.1.1 Tipos de Informes y su Función en las Organizaciones 6](#_Toc176742740)

[1.2 Identificación de tareas en informes 7](#_Toc176742741)

[1.3 Seguimiento automático de tareas 8](#_Toc176742742)

[1.3.1 Aprendizaje automático 9](#_Toc176742743)

[1.3.2 Procesamiento del lenguaje natural 10](#_Toc176742744)

[1.3.3 Soluciones propuestas en la literatura 11](#_Toc176742745)

[1.3.4 Herramientas y tecnologías existentes 13](#_Toc176742746)

[1.4 Conclusiones parciales. 20](#_Toc176742747)

[Capítulo 2: Módulo para la identificación de tareas en documentos de textos 21](#_Toc176742748)

[2.1 Requisitos funcionales 21](#_Toc176742749)

[2.2 Requisitos no funcionales 21](#_Toc176742750)

[2.3 Diagrama de caso de uso del sistema 22](#_Toc176742751)

[2.4 Diagramas de secuencia 22](#_Toc176742752)

[2.4.1 Diagrama de secuencia de Extraer tareas 22](#_Toc176742753)

[2.5 Implementación del módulo de identificación de tareas 23](#_Toc176742754)

[2.6 Conclusiones parciales 25](#_Toc176742755)

[Capítulo 3: Validación 26](#_Toc176742756)

[3.1 Pruebas funcionales 26](#_Toc176742757)

[3.1.1 Preprocesado de datos 26](#_Toc176742758)

[3.1.2 Conformación del Excel 27](#_Toc176742759)

[3.1.3 Extracción de tareas 28](#_Toc176742760)

[3.2 Conclusiones parciales 29](#_Toc176742761)

[Conclusiones generales 31](#_Toc176742762)

[Recomendaciones 32](#_Toc176742763)

[Referencias bibliográficas 33](#_Toc176742764)

[Glosario de términos 39](#_Toc176742765)

Índice de figuras

[Figura 1. Arquitectura de la herramienta propuesta por Kalia et al. para la identificación y seguimiento de compromisos [9]. 12](#_Toc176742766)

[Figura 2. Diagrama de caso de uso del sistema 22](#_Toc176742767)

[Figura 3. Diagrama de secuencia del caso de uso extraer tareas 23](#_Toc176742768)

[Figura 4. Diagrama de clases del módulo 24](#_Toc176742769)

[Figura 5. Entrada de información por consola 26](file:///D:\Tesis\versiones\Ultimas%20versiones\InformePP2_VanesaCasañola.docx#_Toc176742770)

[Figura 6. Texto preprocesado 26](file:///D:\Tesis\versiones\Ultimas%20versiones\InformePP2_VanesaCasañola.docx#_Toc176742771)

[Figura 7. Entrada de información que presenta errores 27](#_Toc176742772)

[Figura 8. Resultado del preprocesado incorrecto 27](#_Toc176742773)

[Figura 9. Conformación del Excel con datos preprocesados 28](#_Toc176742774)

[Figura 10. Entrada para extraer tareas 28](#_Toc176742775)

[Figura 11. Tareas extraídas 29](#_Toc176742776)

[Figura 12. Entrada para extraer tareas con sujeto omitido 29](#_Toc176742777)

[Figura 13. Resultado de la extracción de tareas vacío 29](#_Toc176742778)

Índice de tablas

[Tabla 1 Comparación entre aplicaciones desktop, web y móvil 16](#_Toc176742779)

[Tabla 2 Comparación entre lenguajes de programación 17](#_Toc176742780)

[Tabla 3 Requisitos funcionales del módulo 21](#_Toc176742781)

# Introducción

En las instituciones contemporáneas, los intercambios internos desempeñan un papel fundamental en la resolución de problemas y la toma de decisiones [1]. Estos intercambios, a menudo caracterizados por discusiones cerradas y sensibles, no siempre son accesibles al público general [2]. El contenido de estas interacciones abarca desde la planificación estratégica hasta la resolución de conflictos, y su manejo eficiente es esencial para el funcionamiento óptimo de la organización [3], [4] .

La documentación de estos intercambios a través de informes detallados no solo actúa como un registro oficial, sino que también facilita la comunicación interna y la transparencia [5], [6]. Estos informes reflejan las discusiones, decisiones y acuerdos alcanzados, sirviendo como base para acciones futuras. Sin embargo, dada la complejidad y el volumen de información, la extracción manual de tareas específicas se convierte en un proceso tedioso y susceptible a errores [7], [8].

Identificar tareas dentro de estos informes es un desafío crítico [8], [9]. Las tareas pueden estar implícitas, redactadas de manera ambigua o dispersas a lo largo del documento. Este problema se acentúa cuando la información es de naturaleza sensible, lo que limita el uso de soluciones tecnológicas estándar. La precisión en la identificación y la contextualización de las tareas es crucial para garantizar una correcta ejecución.

Además de la identificación, el seguimiento de las tareas es esencial para asegurar su ejecución y cumplimiento. La falta de seguimiento puede resultar en la pérdida de información importante y en la ineficiencia operativa. Es vital implementar un sistema que no solo identifique las tareas, sino que también asegure que estas se realicen en el tiempo y forma adecuados, facilitando la rendición de cuentas y la evaluación del progreso.

Un desafío adicional es la integración de las tareas identificadas en distintos informes. A menudo, existen tareas repetidas o complementarias que deben unificarse sin perder información crucial. Garantizar que el proceso de integración mantenga la integridad de los datos y evite duplicaciones innecesarias es fundamental para la eficiencia y la coherencia operativa.

En respuesta a estos desafíos, la automatización emerge como una solución efectiva. La aplicación de algoritmos de aprendizaje automático permite identificar tareas de manera automatizada. Esto mejora la precisión y eficiencia del proceso, reduciendo la carga de trabajo y minimizando el margen de error humano [10], [11], [12].

La identificación y seguimiento automáticos de tareas en texto ha sido objeto de varias investigaciones a lo largo de los años, y son muchos los algoritmos y programas desarrollados para dar cumplimento a esto. *Kalia et al.* desarrollaron un sistema para la extracción de tareas y compromisos, utilizando un analizador sintáctico de dependencias para identificar los sujetos, verbos de acción y objetos que deben cumplir con algunas restricciones [9]. Una limitante de este sistema fue que los verbos de acción se identifican a partir de una lista predefinida, sin tener en cuenta el contexto, además que se orienta a correos electrónicos.

*Treude et al.* propusieron un sistema para extraer automáticamente tareas de la documentación de software con el fin de mejorar la navegación y búsqueda de información relevante para los desarrolladores. Identificaron verbos asociados con un objeto directo o una frase preposicional, complementando esta identificación con dependencias gramaticales para verificar las relaciones entre palabras. Sin embargo, una limitante de este enfoque fue la precisión y calidad en la extracción de tareas, ya que el análisis de dependencias gramaticales puede no capturar todas las sutilezas del lenguaje, llevando a errores [13].

En 2020, *Diwanji et al.* introdujeron el modelo Lin, un método no supervisado para identificar tareas en comunicaciones textuales como correos electrónicos y mensajes de chat. Este modelo utiliza reglas que combinan información semántica y sintáctica para extraer tareas, lo que podría ser útil para generar listas de tareas y recordatorios. A pesar de sus avances, se observó que el modelo semántico tendía a producir falsos positivos al no considerar adecuadamente el contexto de los verbos, mientras que el modelo sintáctico, a menudo fallaba en la detección precisa de las tareas [14].

A pesar de los avances tecnológicos, muchas soluciones enfrentan limitaciones significativas, especialmente en entornos donde la información puede ser cerrada y sensible. Las preocupaciones relacionadas con la privacidad, la seguridad y la precisión en la interpretación de contextos restringen la implementación de muchas herramientas disponibles en el mercado, por lo que estas propuestas no están diseñadas para implementarlas en la extracción y seguimiento de tareas en documentos de texto. En este sentido, es esencial desarrollar soluciones adaptadas que respeten las particularidades de estos entornos.

A partir de la **situación problemática** descrita, se identifica como **problema de investigación**: **La limitación** en la disponibilidad de un sistema que permita el análisis de documentos de texto para identificar tareas en estos, así como darles seguimiento de manera automática.

Se señala como **objeto de estudio**, análisis de documentación de intercambios internos. Por lo que el **campo de acción** es la identificación y seguimiento de tareas en documentos de texto utilizando aprendizaje automático y procesamiento del lenguaje natural.

Como propuesta de solución al problema planteado, se define el siguiente **objetivo general**: desarrollar un sistema que pueda identificar y dar seguimiento automático a tareas que se encuentran en documentos de texto. A continuación, para el cumplimiento de este objetivo se han establecido los siguientes **objetivos específicos** con sus correspondientes **tareas** de investigación:

1. Definir la identificación de tareas en documentos y los conceptos de aprendizaje automático y procesamiento del lenguaje natural como una vía de solución.

* Describir los conceptos necesarios para el entendimiento de la problemática. (Práctica)
* Comparar trabajos existentes para la identificación de tareas. (Práctica)
* Comparar tecnologías que puedan utilizarse para el desarrollo del sistema. (Práctica)

1. Desarrollar un sistema para la identificación y seguimiento automático de tareas en documentos.

* Definir requisitos funcionales y no funcionales con los que debe cumplir el sistema. (Práctica y Tesis)
* Diseñar un módulo para la identificación de tareas en documentos de texto. (Práctica)
* Implementación del módulo diseñado. (Práctica)
* Diseñar un sistema que vincule el módulo de extracción de tareas y de seguimiento automático a las tareas extraídas. (Tesis)
* Implementar el sistema diseñado. (Tesis)
* Gestionar la configuración para el funcionamiento adecuado del sistema. (Tesis)

1. Evaluar el sistema

* Diseñar las pruebas y experimentos para comprobar el funcionamiento del módulo de identificación de tareas. (Prácticas)
* Diseñar las pruebas y experimentos para comprobar el funcionamiento del sistema. (Tesis)
* Ejecutar los casos de prueba diseñados. (Prácticas y Tesis)
* Documentar resultados y análisis de los casos de prueba. (Prácticas y Tesis)

Con el desarrollo de este trabajo, se presenta un sistema que sea capaz de identificar tareas en documentos utilizando algoritmos de aprendizaje automático y procesamiento del lenguaje natural y establecer un seguimiento automático de dichas tareas. Por esta razón el **artefacto** de salida se clasifica como un diseño de software.

El **valor práctico** de este trabajo radica en la utilidad que tendría un sistema de identificación y seguimiento automático de tareas en las empresas nacionales. Dotaría a las instituciones de una herramienta que facilitaría un proceso tan tedioso como la extracción manual de tareas, reduciendo en gran medida la carga de trabajo del personal y la posibilidad de errores por parte de los trabajadores. Por otra parte, al contar con un seguimiento automático de tareas, ofrecería un recordatorio para el cumplimiento de estas evitando así, el incumplimiento o la tardanza para realizarlas.

El trabajo se encuentra dividido en introducción, tres capítulos, conclusiones, recomendaciones y referencias bibliográficas. En el **Capítulo 1** se hace referencia a la documentación producida por las instituciones contemporáneas y posibles clasificaciones de estas. Se explica en que consiste una tarea y las vías existentes para el seguimiento automático de las mismas. Se establece una comparación entre los proyectos similares al tema que trata el trabajo y las posibles tecnologías a utilizar para desarrollar el sistema.

En el **Capítulo 2** se recoge una captura de requisitos, funcionales y no funcionales, con los que debe cumplir el sistema que se desea desarrollar. Se hace una propuesta de diseño del sistema que incluye un módulo para la detección de tareas. Además, se define la tecnología seleccionada para implementación.

En el **Capítulo 3** se describe el funcionamiento de las principales funcionalidades del módulo. Se representan varias entradas y los resultados esperados ante cada una. Además, se destacan las desventajas con las que cuenta el módulo y se ponen ejemplos de algunas de ellas y cómo reacciona la implementación ante estas.

# Capítulo 1: Fundamentos teóricos

En este capitulo se hace un estudio del estado del arte, se describen los principales conceptos y términos para el conocimiento del tema. Se indaga en el tema de la documentación en instituciones, que es una tarea y su seguimiento automático. Además, se mencionan trabajos anteriormente realizados que pueden tener relación con el tema que se eta tratando y finalmente se compara entre diferentes tecnologías que pueden ser usadas a lo largo de este trabajo.

## Documentación de intercambio internos en instituciones

En las instituciones contemporáneas los documentos de intercambios internos cubren reuniones-memorandos, documentación de servicios como cronogramas de trabajo, informes de progreso, notas de entrega, cifras y proyecciones de ventas, material educativo y documentos administrativos [15], [16]**.** Esta documentación se refiere a los registros que la organización mantiene y utiliza para informar las decisiones dentro de la empresa constituyendo la base sobre la cual se toman las decisiones [17].

### Tipos de Informes y su Función en las Organizaciones

Las instituciones contemporáneas agrupan gran cantidad de documentación que puede ir desde informes financieros y de productos hasta actas de intercambios internos en grupos de trabajo [18], [19]. Cada registro de documento presenta una particularidad en cuanto a su redacción y a la función que cumplen en las organizaciones.

Los informes financieros se basan en el análisis y la interpretación d datos mediante el uso de un lenguaje matemático. Su función primordial consiste en ayudar a la toma de decisiones de carácter económico, proporcionando información precisa y relevante. Estos documentos permiten a las instituciones evaluar su desempeño y planificar estrategias futuras, consiguiendo una gestión más precisa y eficaz. [18], [20], [21], [22]**.**

Los informes de productos son documentos que describen detalles específicos y características de un bien tangible, permitiendo un seguimiento de este. Este seguimiento ayuda a que las entidades tengan conocimiento sobre el éxito o fracaso de los mismos en el mercado, contribuyendo a realizar inversiones más inteligentes [20], [23], [24].

Los informes de reuniones son documentos que resumen los temas discutidos, acuerdos alcanzados y próximos pasos durante una reunión. También son una vía mediante la cual quedan registrados los planteamientos e inquietudes de los trabajadores, así como recomendaciones de los mismos para el desarrollo de proyectos o perfeccionamiento de algunos, permitiendo dar seguimiento a los acuerdos y evaluar progresos [5], [6], [18].

## Identificación de tareas en informes

Para abordar el tema de la identificación de tareas cabe destacar la denominación que se le da a estas, según lo planteado por *Kalia et al.* se define tarea como una actividad predefinida o creada sobre la marcha por personas en una conversación que contiene un sujeto, un objeto, una acción y un plazo. Si se representa con una T la tarea se define de la siguiente manera: T (sujeto, objeto, acción) [9], [13], [25].

En esta definición el sujeto hace alusión a una entidad de negocio que lleva a cabo la acción. El objeto es la entidad para la que se realiza la acción. Una acción puede ser una disyunción o una conjunción de subacciones. Y finalmente, el plazo establece el tiempo límite para dar cumplimiento a la acción. Un ejemplo de esta situación es cuando Kim envía un mensaje a John diciendo: "Lo recogeré esta noche". Aquí, el sujeto es Kim y el objeto es John, la acción recogerlo y el plazo esta noche. De forma que queda estructurada la tarea de la siguiente manera T (Kim, John, recogerlo) [9].

Por otra parte, *Diwanji et al.* se refieren a tarea como una frase verbal que especifica una única acción a realizar. Consideran que la raíz del verbo de dicha frase verbal es el verbo principal y de esta forma asumen que, a través de la identificación de verbos principales pueden extraer las tareas [14].

Con aumento de la información que se recoge en las reuniones e intercambios que tienen lugar en las instituciones, el análisis y procesamiento de esta se vuelve cada día una labor más compleja [7]**.** Las actas de reuniones y documentos internos a menudo contienen información crucial, pero su correcta comprensión puede resultar difícil debido a la forma en que se encuentran redactados ya sea por su ambigüedad o densidad [9], [26]. La capacidad para extraer los puntos clave y decisiones importantes correctamente es esencial para mantener la eficiencia de la institución.

El análisis de la documentación de forma manual para la extracción de tareas se presenta como una primera vía en este proceso, pero puede estar sujetos a errores debido a la subjetividad en la interpretación de las promesas o criterios recogidos en el texto [26]. Los humanos pueden tener diferentes opiniones para identificar una tarea, lo que lleva a inconsistencias en los acuerdos que se toman y la forma en que se ejecutan [7], [9]. Cuando se trata de textos breves y un contexto limitado, la comprensión de estos se vuelve un proceso inherentemente complejo para las personas [8],limitando la correcta extracción de las tareas.

La documentación de intercambios internos generada por las empresas va en constante aumento por lo cual, cada vez es mayor el número de actas y documentos a analizar en busca de tareas que surgen [9]. Este proceso puede resultar agotador para los trabajadores, produciendo cansancio y fatiga al realizar por un tiempo prolongado esta labor. Teniendo en cuenta esto, se produce un retraso en la finalización de los procesos de extracción de tareas consumiendo gran cantidad de tiempo hasta su finalización.

Las barreras de la comunicación dentro de las organizaciones pueden resultar ser sutiles pero significativas. La terminología, la jerga y las diferencias entre los estilos de comunicación de los departamentos puede llevar a malentendidos. la precisión en la comunicación y escritura de los documentos es vital para superar estos desafíos.

## Seguimiento automático de tareas

Los sistemas automáticos de gestión de tareas pueden superar los problemas de detección y seguimiento de las mismas y ayudar a las personas a gestionar estas de manera más eficiente [7], [27]. Detectar tareas automáticamente permite que las herramientas de gestión de tareas y los asistentes digitales generen recordatorios y notificaciones para secundar a los usuarios a cumplir con sus obligaciones [7].

### Aprendizaje automático

Una alternativa para la automatización de los procesos clásicos es el aprendizaje automático (*Machine Learning*, ML por sus siglas en inglés), una subdisciplina de la inteligencia artificial que utiliza algoritmos entrenados con conjuntos de datos para crear modelos que permiten a las máquinas realizar tareas tradicionalmente hechas por humanos, como categorizar imágenes, analizar datos o predecir fluctuaciones de precios. Estos algoritmos detectan patrones y aprenden a hacer predicciones y recomendaciones procesando datos y experiencias, adaptándose y mejorando con el tiempo a medida que reciben nueva información [28], [29], [30].

El ML cuenta con diferentes enfoques de trabajo que se dividen tradicionalmente en tres categorías amplias, que corresponden a paradigmas de aprendizaje, dependiendo de la naturaleza de la "señal" o "retroalimentación" disponible para el sistema de aprendizaje:

**Aprendizaje supervisado:** El aprendizaje supervisado implica presentar ejemplos de entradas y sus salidas deseadas a la computadora para que aprenda a asociar entradas con salidas. Los algoritmos construyen un modelo matemático utilizando datos de entrenamiento, representados como vectores de características y matrices. Mediante la optimización iterativa, se aprende una función que predice las salidas de nuevas entradas. Si el algoritmo mejora sus predicciones con el tiempo, se considera que ha aprendido [31]. Entre los tipos de algoritmos supervisados se incluyen el aprendizaje activo, la clasificación y la regresión [28].

**Aprendizaje no supervisado:** El aprendizaje no supervisado implica que el algoritmo encuentre estructuras en los datos sin etiquetas ni clasificación previa. Se enfoca en descubrir patrones ocultos o en el aprendizaje de características. Los algoritmos identifican similitudes en los datos y reaccionan según la presencia o ausencia de estas similitudes [31]. Sus aplicaciones principales incluyen agrupamiento, reducción de dimensionalidad y estimación de densidad [32].

**Aprendizaje por refuerzo:** El aprendizaje por refuerzo involucra a un programa que interactúa con un entorno dinámico para alcanzar un objetivo, recibiendo retroalimentación en forma de recompensas que busca maximizar. Este campo estudia cómo los agentes de software deben tomar acciones en un entorno para maximizar recompensas acumulativas [31]. Se aplica en diversas disciplinas como teoría de juegos, teoría de control, investigación operativa, teoría de la información, estadísticas y algoritmos genéticos [33].

### Procesamiento del lenguaje natural

Otro procedimiento que se utiliza para la automación de procesos es el procesamiento del lenguaje natural (*Natural Lenguaje Processing*, NLP por sus siglas en inglés), otra rama de la inteligencia artificial que integra elementos de varias áreas, como la ciencia de la computación y la lingüística computacional, con el objetivo de mejorar la comunicación entre los humanos y las máquinas [34]. Su propósito es dotar a las computadoras de la capacidad de comprender, interpretar y generar expresiones humanas de manera natural, facilitando la comprensión del lenguaje por parte de las máquinas para que se asemeje más a la manera en que las personas lo entienden [35].

El NLP ha evolucionado drásticamente en las últimas décadas. Los sistemas basados en reglas dominaron la investigación temprana, dependiendo de reglas y léxicos elaborados manualmente para analizar estructuras lingüísticas [36]. Sin embargo, estos enfoques eran limitados en cobertura y específicos de dominio. El NLP estadístico surgió en la década de 1990, utilizando el aprendizaje automático para inducir reglas y patrones a partir de grandes corpus de texto [37]. Métodos como los modelos de lenguaje n-gram, los clasificadores de Bayes ingenuos y las máquinas de soporte vectorial se volvieron populares para tareas del NLP [38].

Entre las ventajas que ofrece se encuentra la capacidad de automatizar tareas que implican el procesamiento de texto, incluyendo la extracción de información relevante y la clasificación de documentos, lo que mejora significativamente la eficiencia y productividad en diversas áreas [35]. Esto optimiza procesos internos y libera tiempo y recursos para otras actividades, aumentando la productividad general. Además, puede analizar grandes volúmenes de datos no estructurados, desde redes sociales hasta correos electrónicos y documentos, permitiendo la extracción de información valiosa y el descubrimiento de patrones y tendencias que podrían pasar desapercibidos para el ojo humano [39]. Esta capacidad de análisis abre nuevas posibilidades en campos como el marketing, la investigación social y la inteligencia empresarial.

### Soluciones propuestas en la literatura

El área de la automatización de procesos como la extracción de tareas y seguimiento de estas ha sido objeto de estudio en los últimos años destacándose investigaciones como la realizada por *Kalia et al.* en 2013 donde desarrollan un sistema para la clasificación y seguimiento de compromisos en correos electrónicos y chat, en la ***Figura 1*** se puede observar la arquitectura que sigue esta herramienta que ofrece un agente que se puede conectar a mensajería de chat en línea e identifica compromisos en un panel para verificación y confirmación de los participantes en la conversación. Cuenta con un analizador de oraciones que extrae detalles y características potenciales de la tarea, utiliza un componente predictor y un modelo entrenado que identifica la clase de las características extraídas. Por último, cuenta con un componente visual para reflejar el resumen de tareas y compromisos basados en los chats [9].

A pesar de la utilidad que puede tener esta investigación presenta ciertas limitantes como son: la utilización de una lista de verbos de acción predefinidos por lo cual en el momento de identificar estos no se tiene en cuenta el contexto. Para identificar una estructura de tarea depende de la característica de dependencia escrita que no trabaja con precisión cuando las oraciones son demasiado largas o muy cortas. Según la definición de tarea y compromiso que utilizan, si no encuentra definido claramente quien es el deudor y acreedor de determinada sentencia o sujeto y objeto, no se pueden identificar correctamente las tareas y compromisos que surgen.

Figura 1 


Figura 1. Arquitectura de la herramienta propuesta por Kalia et al. para la identificación y seguimiento de compromisos [9].

Otra investigación de interés fue la realizada en 2015 por *Treude et al.* donde se centran en la extracción automática de tareas de desarrollo a partir de la documentación de software, con el objetivo de facilitar la navegación y búsqueda de información relevante para los desarrolladores. La extracción de tareas se lleva a cabo utilizando un enfoque de procesamiento del lenguaje natural y análisis sintácticos. El proceso de extracción se basa en la identificación de verbos, pero a diferencia del trabajo realizado por *Kalia et al.*, este busca los que están asociados con un objeto directo y/o una frase preposicional, para complementar este paso se utilizan las dependencias gramaticales para verificar la relación que tienen las palabras entre sí. Esto es crucial para identificar correctamente las tareas, ya que las relaciones entre verbos, objetos y preposiciones pueden variar en complejidad [13].

A pesar de lo novedoso de esta investigación para la extracción automática de tareas de tareas, se identifican limitantes en cuanto a la precisión y calidad de las tareas encontradas pues el análisis de dependencias gramaticales puede no capturar todas las sutilezas del lenguaje natural, lo que podría llevar a errores en la identificación de tareas. Además, el sistema de *TaskNavigator* que proponen carece de técnicas de aprendizaje automático, lo que significa que no puede adaptarse o mejorar automáticamente con el tiempo a partir de nuevos datos o retroalimentación de los usuarios. Esto puede limitar su capacidad para evolucionar y mejorar su precisión en la extracción de tareas.

En 2020, *Diwanji et al.* desarrollaron el enfoque Lin, que es un método no supervisado para identificar tareas específicas en la comunicación textual, como correos electrónicos y mensajes de chat. Este extrae las tareas de las comunicaciones textuales utilizando un enfoque basado en reglas que combinan la información semántica y sintáctica, si se realiza de forma precisa puede ser beneficiosa para la generación de listas de tareas, recordatorios y entradas en calendarios [14].

Entre limitantes que presenta este trabajo se destaca que el modelo semántico de Lin puede generar falsos positivos al considerar verbos sin tener en cuenta el contexto en el que aparecen, resultando en la identificación incorrecta de múltiples tareas en oraciones donde solo hay una. También, el modelo de sintaxis obtiene un bajo nivel de *recall*, existiendo la probabilidad de marcar verbos incorrectos como tareas y no considera sus descendientes, que podrían incluir las tareas correctas, aumentando el valor de los falsos negativos.

### Herramientas y tecnologías existentes

Actualmente son diversas las tecnologías que se utilizan para el diseño y realización de un sistema de gestión. Por tanto, es necesario realizar un análisis de las herramientas a considerar para el desarrollo a partir de las tendencias y tecnologías actuales, en este epígrafe se propone un análisis de dichas tecnologías y herramientas que pueden ser útiles para el desarrollo de este trabajo.

#### Lenguaje de modelado

**Lenguaje unificado de modelado** (*Unified Modeling Language*, UML por sus siglas en inglés): Es un lenguaje gráfico que se utiliza para documentar, construir, visualizar y especificar los artefactos de un sistema de software. Incluye elementos específicos, como esquemas de bases de datos y expresiones de lenguajes de programación, para describir un plano del sistema modelo [40].

Es crucial destacar que UML es un “lenguaje de modelado” para describir o especificar métodos o procesos. Se emplea para establecer un sistema, documentar y construir, así como para detallar los artefactos del sistema. En otras palabras, es el idioma en el que se explica el modelo [41].

#### Herramienta Ingeniería de software asistida por computadoras (CASE)

**Rational Rose**

*Rational* es una herramienta de Ingeniería Asistida por Computadora (*Computer Aided Software Engineering*, CASEpor sus siglas en inglés) desarrollada por *Rational Corporation* basada en UML que permite crear los diagramas que se van generando durante el proceso de Ingeniería en el desarrollo del software. Como tiene una amplia gama de estereotipos predefinidos que facilitan el proceso de modelación del software, facilita significativamente la creación de documentación del software que se está desarrollando [42].

Características [42], [43]:

* Permite Especificar, Analizar y Diseñar el sistema antes de Codificarlo.
* Mantiene la consistencia de los modelos del sistema software.
* Chequeo de la sintaxis UML.
* Generación de la documentación automáticamente.
* Generación de Código a partir de los Modelos.
* Ingeniería Inversa.

**Visual Paradigm**

Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, prueba y despliegue [44].

Este sigue el enfoque de Arquitectura Dirigida por Modelos (*Model-Driven Architecture*, MDA por sus siglas en inglés). La transformación de modelos que provienen del espacio del problema hasta modelos específicos de la plataforma es el proceso central, y sus principios son la estandarización, la abstracción y la automatización [44], [45].

Características [44], [45], [46]:

* Ingeniería inversa.
* Generación de código.
* Editor de Detalles de Casos de Uso.
* Diagramas de flujo de datos.
* Soporte mapeo objeto-relacional (ORM).
* Generación de Base de Datos (BD).
* Generador de informes para generación de documentación.

#### Aplicaciones en plataformas digitales

Al desarrollar un proyecto es indispensable escoger la tecnología a utilizar. Debido a las múltiples opciones que existen en la actualidad, esta selección resulta compleja, por lo que es necesario tener en cuenta las necesidades del sistema para realizar una selección ajustada y que cumpla con las expectativas. Al comparar las aplicaciones de **escritorio**, ***web*** y**móviles**se comprobó que ambas poseen sus particularidades.

Las aplicaciones de **escritorio** se integran completamente con el sistema operativo y el hardware de la computadora, lo que les permite acceder a los periféricos y aprovechar al máximo la configuración instalada. Como resultado, tienen un mejor tiempo de respuesta de la interfaz, pero requieren de instalación. [47]. Se crean normalmente para un sistema operativo particular y, por lo general, requieren ser actualizadas en cada cliente. El rendimiento de la aplicación también puede depender del hardware que esté instalado [48].

Las aplicaciones ***web***, sin embargo, pueden ser accedidas desde cualquier sitio con el uso de un navegador de Internet. En la mayoría de los casos no es necesario realizar actualizaciones en los clientes, debido a que estas se realizan únicamente en el servidor. Esto elimina los problemas de incompatibilidad entre versiones, ya que todos los usuarios trabajan con la misma y los requisitos del lado del cliente son mínimos y suelen estar incluidos en el paquete del sistema operativo [49]. A pesar de que podrían existir diferencias en la interfaz a raíz del navegador *web* o dificultades de integración con otro sistema aparte del navegador web, agregar un nuevo puesto de trabajo solo requiere la instalación de una computadora nueva y no requiere un sistema operativo específico [48].

Las aplicaciones **móviles** por su parte están diseñadas para ejecutarse en un dispositivo móvil, como un teléfono inteligente o una *tablet*. A diferencia de las aplicaciones de **escritorio** o ***web***, las aplicaciones móviles están diseñadas para aprovechar el alcance, la portabilidad, las interfaces de pantalla táctil y las diversas funcionalidades sin precedentes de las plataformas móviles [50], [51].

Tabla 1 Comparación entre aplicaciones desktop, web y móvil

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterios** | **Aplicación de escritorio** | **Aplicación web** | **Aplicación móvil** |
| **Portabilidad** | Limitada, se utiliza solo en el ambiente donde se instaló | Se puede acceder desde cualquier dispositivo con acceso a Internet | Limitada a una plataforma específica |
| **Actualización** | Requiere de actualizaciones manuales en todas las terminales donde está instalado el *software* | Se realizan de forma automática y simultánea en todos los equipos desde una única ubicación | Requiere de actualizaciones manuales |
| **Instalación** | Requiere instalación | No requiere instalación | Requiere instalación |
| **Conexión a Internet** | Se requiere en dependencia de las características del *software* | Siempre requiere conexión a Internet para su uso | Accesible sin conexión en dependencia de la configuración del *software* |

#### Lenguajes de programación

Entre los lenguajes de programación existentes más populares para el desarrollo *backend* se encuentra ***Python*** un lenguaje interpretado, orientado a objetos, de alto nivel y con semántica dinámica [52]. Posee una la sintaxis simple y fácil de entender, enfatiza la legibilidad y reduce los costos de mantenimiento del programa. Promueve la reutilización y modularización del código mediante soporte para módulos y paquetes. La extensa biblioteca estándar de *Python* y su intérprete están disponibles gratuitamente en formato binario o fuente en todas las plataformas principales [52], [53].

Por otra parte, ***Java*** es un lenguaje de propósito general, orientado a objeto, multiplataforma y compilado [54]. Es apreciado por su simplicidad en la sintaxis, su soporte para multihilos, su enfoque en la seguridad, y su gran comunidad global, lo que lo convierte en una opción versátil y robusta para el desarrollo de aplicaciones web, móviles y empresariales [54], [55].

C++ es otro lenguaje de programación de alto nivel que fue desarrollado como una extensión del lenguaje C. Es conocido por su capacidad de soportar tanto programación orientada a objetos como programación procedural y genérica, lo que lo convierte en una herramienta versátil para crear una amplia variedad de aplicaciones [56]. Ofrece a los programadores un alto grado de control sobre los recursos del sistema y la memoria, lo que lo hace ideal para proyectos que requieren eficiencia y rendimiento [57].

Tabla 2 Comparación entre lenguajes de programación

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterios** | **Python** | **Java** | **C++** |
| Orientado a objeto | Si | Si | Si |
| Multiplataforma | Si | Si | Si |
| Vinculación con herramientas de NLP | Existen herramientas y se mantienen actualizadas | Existen herramientas y se mantienen actualizadas | Existen herramientas, pero muchas llevan un prolongado tiempo sin actualizaciones |
| Curva de aprendizaje | Baja | Baja | Alta |

#### Herramientas de procesamiento de lenguaje natural

**NLTK** está basada en *Python*. Ofrece interfaces fáciles de usar para más de 50 corpus y recursos léxicos, además de una variedad de bibliotecas para clasificación, tokenización, etiquetado, análisis sintáctico y razonamiento semántico. También incluye bibliotecas de procesamiento de lenguaje natural de nivel industrial y cuenta con una comunidad activa. es gratuito, de código abierto y multiplataforma [58].

**Spacy** es una biblioteca de código abierto en *Python* para procesamiento del lenguaje natural. Se emplea en tareas como análisis de textos, extracción de información, y comprensión del lenguaje natural. Esta facilita la tokenización, el reconocimiento de entidades nombradas, el etiquetado de partes del discurso y el análisis de dependencias en textos [59]. Los modelos que brinda están disponibles para varios idiomas, incluyendo inglés, español, francés, alemán, chino, entre otros [60].

**Stanford CoreNLP** es una herramienta de procesamiento de lenguaje natural de código abierto diseñada en Java para proporcionar un análisis robusto de textos, incluye tokenización, análisis sintáctico, reconocimiento de entidades nombradas, y resolución de correferencias, entre otros. Posee modelos diversos en idiomas como el inglés, francés, alemán, chino y árabe principalmente. Proporciona salidas de información en formatos como JSON y XML [61].

**Freeling** es una herramienta de análisis lingüístico de código abierto, desarrollada en C++. Ofrece funciones avanzadas como análisis morfológico, detección de entidades nombradas, etiquetado PoS, análisis sintáctico, desambiguación semántica, etiquetado de roles semánticos, entre otros. Cuenta con diversos idiomas como inglés, español, portugués, italiano, francés y más. Además, incluye una interfaz de línea de comandos para analizar textos y exportar los resultados en formatos como XML, JSON, y CoNLL [62].

#### Herramientas para el desarrollo de frontend.

**Angular**

Angular es un *framework* de *JavaScript* de código abierto desarrollado por *Google* para la creación de aplicaciones *web* de página única (SPA) y aplicaciones *web* dinámicas. Se desarrolla usando *TypeScript*, un superconjunto de *JavaScript* que añade características como tipado estático y decoradores. Implementa patrones arquitectónicos como Modelo-Vista-Controlador (MVC), lo que facilita la gestión de aplicaciones complejas al separar datos, vistas y controladores. Además, posee un sistema de enlace de datos bidireccional que permite reflejar automáticamente en la interfaz cambios en el diseño. Es compatible con todos los navegadores *web* modernos, incluyendo los móviles, y es ideal para construir aplicaciones robustas y escalables [63].

**Vue.js**

*Vue.js* comúnmente conocido como *Vue*, es un *framework* perteneciente a *JavaScript* de código abierto que se utiliza principalmente para crear interfaces de usuario. Cuenta con patrones arquitectónicos como Modelo-Vista-Vista-Modelo (MVVM) [64]. Es progresivo, lo que significa que es accesible, adaptable y se puede integrar fácilmente con otros proyectos o bibliotecas existentes. Se destaca por su eficiencia y rendimiento, además de poseer una sintaxis simple y una API flexible. Entre sus características se incluye la implementación rápida del DOM virtual, ligereza y enlace de datos bidireccional [64], [65].

**React**

*React* es una biblioteca de *JavaScript* de código abierto, desarrollada por Facebook, tiene el objetivo de simplificar el complejo proceso de desarrollo de interfaces de usuario interactivas y dinámicas. La creación de componentes reutilizables que actúan como bloques independientes, permitiendo la construcción modular y eficiente de aplicaciones [66]**.** Entre sus características se destaca la presencia de un DOM virtual, creación de aplicaciones de página única donde se actualizan dinámicamente las secciones del contenido sin necesidad de recargar la página completa y patrones arquitectónicos como Modelo-Vista-Vista-Modelo (MVVM) que se enfoca en la capa vista permitiendo una separación clara entre los datos, la lógica y la presentación, haciendo más fácil el manejo y mantenimiento de aplicaciones complejas [67].

## Conclusiones parciales.

Con la realización de este capitulo se da cumplimiento al objetivo de realizar un análisis del estado del arte con relación al tema que se trata:

* Se describe que son los documentos de institucionales y que como se pueden clasificar.
* Se aborda sobre que es una tarea y las posibles formas de identificarlas.
* Se describen soluciones propuestas en la literatura por varios autores que trataron sobre el tema.
* Se compara tecnologías y herramientas a utilizar en el desarrollo de este trabajo como son los lenguajes de programación, herramientas para procesamiento de lenguaje natural, entre otras.

# Capítulo 2: Módulo para la identificación de tareas en documentos de textos

El siguiente capítulo propone un módulo de extracción de tareas en documentos. Se describen los requisitos tanto funcionales como no funcionales con los que debe contar el módulo que se implementa. Además, se representan las funcionalidades con las que cuenta el módulo propuesto a través de diagramas de casos de uso del sistema y diagramas de secuencias. Finalmente, se expone la estructura del modulo y se describe su funcionalidad.

## Requisitos funcionales

El módulo que se implementa debe cumplir con determinados requisitos funcionales para el correcto funcionamiento del mismo. Entre estos se encuentran los recogidos en la siguiente tabla:

Tabla 3 Requisitos funcionales del módulo

|  |  |
| --- | --- |
| NO. | Requisitos |
| 1 | El módulo debe reconocer la información que se le envía |
| 2 | El módulo debe ser capaz preprocesar la información introducida. |
| 3 | El módulo debe permitir la exportación del resultado obtenido en formato CSV o XLSX. |

## Requisitos no funcionales

El módulo debe cumplir con los requisitos no funcionales que se recogen en la siguiente tabla:

|  |  |
| --- | --- |
| NO. | Requisitos |
| 1 | Usabilidad: los componentes del modulo deben cumplir con sus funcionalidades correctamente. |
| 2 | Fiabilidad: debe informársele al usuario en todo momento el estado actual de la operación que está realizando (existe algún error, está siendo procesada o culminó satisfactoriamente). |

## Diagrama de caso de uso del sistema

El actor del negocio, en este caso el analista, se encarga de ejecutar la funcionalidad de extraer tareas. Este proceso se realiza a través de la inserción de la información que desea procesar para la extracción de las tareas. Una vez introducida la información mediante texto plano, se preprocesan los datos y se procede a extraer las tareas existentes dentro de la información enviada. Al concluir este proceso se exporta un archivo en formato CSV con la información resultante. En la *Figura 2* se puede observar un diseño del diagrama de casos de uso del sistema que responde a lo anteriormente descrito.

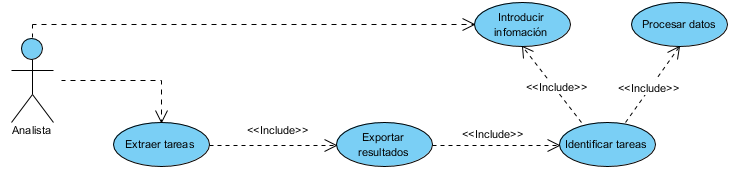


Figura 2. Diagrama de caso de uso del sistema

## Diagramas de secuencia

En este epígrafe se presentan diagramas de secuencias que describen el funcionamiento del caso de uso relacionados con el actor del negocio, Extraer tareas.

### Diagrama de secuencia de Extraer tareas

Como se representa en la *Figura 3* cuando el analista introduce la información que desea analizar mediante texto plano se envía al módulo desarrollado donde se preprocesa la información y se ejecuta la extracción de tareas en este. En caso de que todo se desarrolle de manera factible, se devuelve un documento de extensión CSV O XLSX que contiene todas las tareas identificadas en la información proporcionada. Por otra parte, en caso de ocurra algún error en el proceso que puede estar dado por una mala escritura en formato plano, se notifica al usuario.

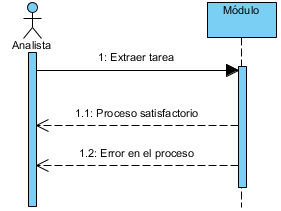


Figura 3. Diagrama de secuencia del caso de uso extraer tareas

## Implementación del módulo de identificación de tareas

El módulo va orientado a la plataforma web. Este será integrado con un sistema que se desarrollará en futuros trabajos y proporcionará una interfaz de usuario. Para la realización de este módulo se utilizó Python como lenguaje de programación por las características descritas en sección 1.3.4 del capitulo anterior y spacy como herramienta para el procesamiento del lenguaje natural, de igual forma la justificación de su selección se puede encontrar al analizar las características que este presenta en el epígrafe 1.3.4.

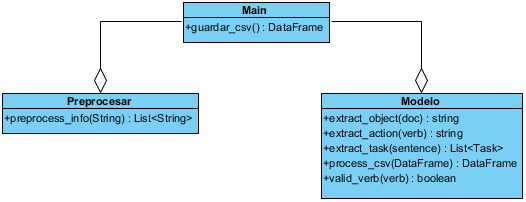


Figura 4. Diagrama de clases del módulo

En la Figura 4 se puede observar la distribución de las clases que sigue el modulo implementado. En clase Main se inica el proceso, esta da lugar a que se recoja la información que se desea analizar, una vez se tiene se invoca la función **preprocess\_info**, esta función recibe por parámetros la información que se introdujo anteriormente y se encarga de eliminar caracteres especiales, espacios en blanco y segmentar la información en oraciones.

Una vez concluido este proceso se procede a guardar los datos resultantes del preprocesado en un archivo CSV, a través de la función **guardar\_csv** que se encuentra en el main. Seguido de esto se invoca la función **process\_csv** perteneciente a la clase modelo, esta se encarga de leer el csv que recibe por parámetro y recorrer cada fila de este mientras la función de extraer tareas se va ejecutando con cada sentencia hasta que finalmente termina el ciclo y se arma nuevamente el csv con todas las sentencias que contiene y agregando los resultados del proceso de extracción de tareas.

La función **extract\_task** es la encargada de identificar las tareas que existen en las sentencias que se le van pasando como se explicó anteriormente, esta primeramente carga las sentencias que se le pasan con el modelo mediano en español de spacy para el procesamiento del lenguaje natural. Después, procede a buscar el sujeto de la sentencia si se encuentra se procede a validar el verbo para garantizar que sea un verbo de acción. A continuación, se ejecutan las funciones **extract\_object** que se encarga de buscar el objeto al cual esta relacionado la tarea y **extract\_action** que devuelve el lemma o infinitivo de dicho verbo. Se procede a construir la tarea poniendo sujeto y acción y en caso de que tenga el objeto, acto seguido se rellena una lista que contiene sujeto, acción, objeto y tarea de la sentencia analizada.

## Conclusiones parciales

Con el desarrollo de este capitulo se da cumplimiento a los objetivos de diseñar e implementar un módulo para la detección de tareas en documentos de texto:

* Se definen los requisitos funcionales y no funcionales del módulo.
* A través del diagrama de casos de uso del sistema se da a entender cómo funciona el módulo.
* El diagrama de secuencia describe la comunicación que existe entre los usuarios y el módulo.
* El diagrama de clases ofrece una vía para entender el funcionamiento de cada método existente en el módulo y cómo se establece la comunicación entre ellos.

# Capítulo 3: Validación

En este capítulo se realizan pruebas funcionales sobre el módulo de extracción de tareas desarrollado. Se comprueba el funcionamiento las principales funcionalidades que incorpora el módulo y se identifican posibles mejoras a realizar.

## Pruebas funcionales

En este epígrafe se desarrollan pruebas funcionales sobre el módulo, se evalúa el correcto funcionamiento del mismo y las mejoras a realizar a partir de los resultados obtenidos.

### Preprocesado de datos

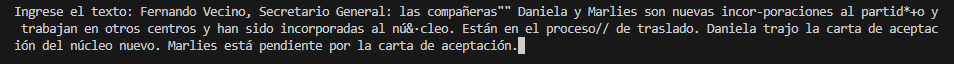
Esta funcionalidad consiste en eliminar caracteres especiales y signos que se encuentren en la información que se introduce, además de separar en oraciones esta para su posterior utilización y lleva todo a letra minúscula. En la *Figura 5* se puede observar la entrada que se realiza a través de la consola.

Figura 5. Entrada de información por consola

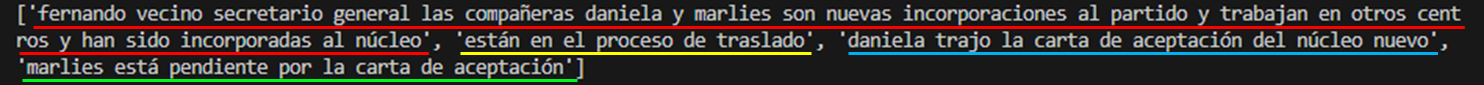
Una vez preprocesado los datos, la información resultante se muestra en la *Figura 6,* aquí se ve como fueron eliminados los símbolos y se convierte a letra minúscula haciendo legible el texto en las diferentes oraciones que lo conformaban.

Figura 6. Texto preprocesado

Las oraciones se apilan en una lista como resultado de este proceso, en la *Figura 6* también se destacan en diferentes colores cada una de las oraciones por las cuales estaba formado el texto que se introdujo para preprocesar.

Un problema con el que cuenta este preprocesado de datos es que, al dividir la información en oraciones, en caso de que se coloque equivocadamente en el momento de escribir un punto se dé un espacio este lo reconoce automáticamente como otra oración. Ejemplo de esto se ve reflejado en la *Figura 7*, que sería la entrada que se hace por consola que presenta el error.

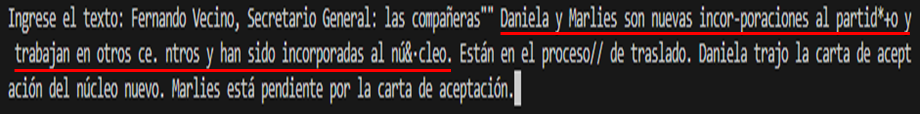




Figura 7. Entrada de información que presenta errores

Al preprocesar esta información vemos reflejado en la *Figura 8* como la oración que aparece subrayada en la *Figura 7* se divide en dos y pierde su significado.

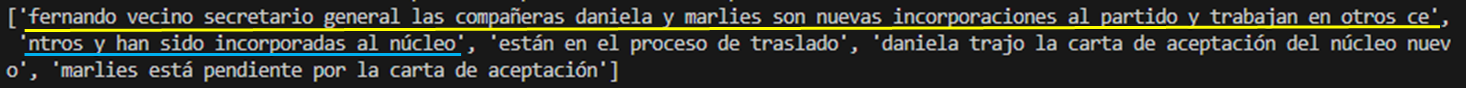


Figura 8. Resultado del preprocesado incorrecto

### Conformación del Excel

El módulo desarrollado trabaja con un archivo en formato xlsx que inicialmente está formado por todas las oraciones que conforman la información introducida. Como se puede ver en la *Figura 7* teniendo en cuenta el texto preprocesado anteriormente el documento Excel quedaría conformada de la siguiente manera.

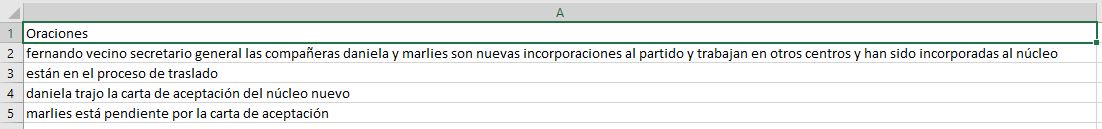


Figura 9. Conformación del Excel con datos preprocesados

El Excel conformado con los datos preprocesados contiene una etiqueta en la cabecera denominada **Oraciones** que es la que se utiliza como guía cuando se identifican las tareas que se encuentran en dicha información.

### Extracción de tareas

La funcionalidad de extracción de tareas parte de la información que está contenida en el documento Excel previamente analizado, entre los posibles resultados esperados para esta información que se presenta en la *Figura 10* son los siguientes:

* Círculo político realizar en teatro.
* Marlies traer carta.
* Rogej representar núcleo.

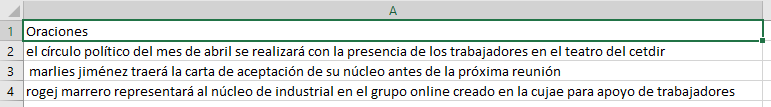


Figura 10. Entrada para extraer tareas

En la Figura 11 se ilustra el resultado del análisis de extracción de tareas y se ve como en algunos casos como el de la primera oración el modulo no es capaz de detectar cual sería el objeto de dicha oración.

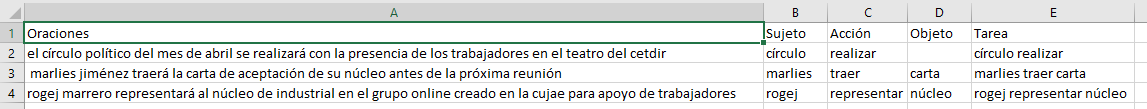


Figura 11. Tareas extraídas

Otro error con el que cuenta este módulo viene ligado a la detección de tareas cuando esta presenta un sujeto omitido como se representa en la Figura 12.

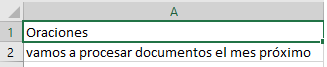


Figura 12. Entrada para extraer tareas con sujeto omitido

Al pasarle esta sentencia al módulo de deteccion de tareas se espera el siguiente resultado:

* Nosotros procesar documentos próximo mes.

Como aparece reflejado en la Figura 13 el resultado que se obtiene es vacío, es decir que, el módulo no encuentra ninguna tarea en la entrada proporcionada esto como anteriormente se planteó se debe a que el sujeto de la oración se encuentra omitido. El reconocimiento de tareas está diseñado para que, primeramente, busque el sujeto de la oración y a partir de él, identifique las demás partes necesarias para completar la tarea. En caso de no encontrar sujeto se concluye que no existe tarea en la información proporcionada.

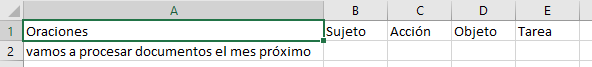


Figura 13. Resultado de la extracción de tareas vacío

## Conclusiones parciales

Con el desarrollo de este capítulo se cumple con el objetivo de evaluar el módulo de extracción de tareas en documentos de texto:

* Se comprueba el funcionamiento de las funcionalidades que integran dicho modulo.
* Se destacan algunos errores que pueden estar presentes a la hora de manejar cierta información.

# Conclusiones generales

Al término del presente trabajo, se llegan a las siguientes conclusiones:

* La automatización del proceso de extracción y seguimiento de tareas en documentos es necesario debido a la gran cantidad de información que se maneja actualmente.
* La utilización de procesamiento del lenguaje natural para identificación de tareas resulta muy ventajosa.
* La utilización del modelo mediano que proporciona Spacy obtiene resultados buenos en la identificación de tareas.

# Recomendaciones

Al concluir este trabajo, se tienen como recomendaciones las siguientes:

* Desarrollar un módulo para el seguimiento de tareas que se vincule al diseñado.
* Desarrollar un sistema que abarque el proceso de identificación y seguimiento automático de tareas.
* Perfeccionar el modulo de deteccion de tareas para garantizar que se subsanen los errores que presenta actualmente.

# Referencias bibliográficas

[1] G. Bucăţa y M. Rizescu, «The Role of Communication in Enhancing Work Effectiveness of an Organization», *Land Forces Acad. Rev.*, vol. 22, mar. 2017, doi: 10.1515/raft-2017-0008.

[2] R. Ducato, «Data protection, scientific research, and the role of information», *Comput. Law Secur. Rev.*, vol. 37, p. 105412, jul. 2020, doi: 10.1016/j.clsr.2020.105412.

[3] D. Mandel Gandrita, «Improving Strategic Planning: The Crucial Role of Enhancing Relationships between Management Levels», *Adm. Sci.*, vol. 13, p. 211, sep. 2023, doi: 10.3390/admsci13100211.

[4] M. Iyamabhor, F. Ndudi, J. Ogundare, y K. Ofune, «Perspective in workplace conflict management strategies and organizational performance: A review of the extant literature», vol. 15, pp. 68-85, mar. 2024.

[5] K. Mishra, L. Boynton, y A. Mishra, «Driving Employee Engagement: The Expanded Role of Internal Communications», *Int. J. Bus. Commun.*, vol. 51, pp. 183-202, abr. 2014, doi: 10.1177/2329488414525399.

[6] A. Schnackenberg y E. Tomlinson, «Organizational Transparency: A New Perspective on Managing Trust in Organization-Stakeholder Relationships», *J. Manag.*, vol. 42, mar. 2014, doi: 10.1177/0149206314525202.

[7] H. Azarbonyad, R. Sim, y R. W. White, «Domain Adaptation for Commitment Detection in Email», en *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, en WSDM ’19. New York, NY, USA: Association for Computing Machinery, ene. 2019, pp. 672-680. doi: 10.1145/3289600.3290984.

[8] M. Sappelli, G. Pasi, S. Verberne, M. de Boer, y W. Kraaij, «Assessing e-mail intent and tasks in e-mail messages», *Inf. Sci.*, vol. 358-359, pp. 1-17, sep. 2016, doi: 10.1016/j.ins.2016.03.002.

[9] A. Kalia, H. R. Motahari Nezhad, C. Bartolini, y M. Singh, «Identifying business tasks and commitments from email and chat conversations», ene. 2013.

[10] N. R. Vajjhala y K. D. Strang, «An Exploratory Big Data Approach to Understanding Commitment in Projects», en *Good Practices and New Perspectives in Information Systems and Technologies*, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, y A. Poniszewska-Marańda, Eds., Cham: Springer Nature Switzerland, 2024, pp. 66-75. doi: 10.1007/978-3-031-60227-6\_6.

[11] W.-J. Kim, J.-H. Byeon, y Y.-J. Kwon, «Development of An Inventory to Classify Task Commitment Type in Science Learning and Its Application to Classify Students’ Types», *J. Korean Assoc. Res. Sci. Educ.*, vol. 33, may 2013, doi: 10.14697/jkase.2013.33.3.679.

[12] A. Kalia, Z. Zhang, y M. P. Singh, «Trustworthy Decision Making via Commitments ?», 2013. Accedido: 24 de junio de 2024. [En línea]. Disponible en: https://www.semanticscholar.org/paper/Trustworthy-Decision-Making-via-Commitments-Kalia-Zhang/fb26bf96c28d9f67e6acd4e5cbaa3886a3d9d284

[13] C. Treude, M. P. Robillard, y B. Dagenais, «Extracting Development Tasks to Navigate Software Documentation», *IEEE Trans. Softw. Eng.*, vol. 41, n.o 6, pp. 565-581, jun. 2015, doi: 10.1109/TSE.2014.2387172.

[14] P. Diwanji, H. Guo, A. Kalia, y M. Singh, «Liṅ: Unsupervised Extraction of Tasks from Textual Communication», dic. 2020. doi: 10.18653/v1/2020.coling-main.164.

[15] P. Năstase, D. Stoica, F. Mihai, y A. Stanciu, «FROM DOCUMENT MANAGEMENT TO KNOWLEDGE MANAGEMENT», 2009.

[16] N. Kameo y J. Whalen, «Organizing Documents: Standard Forms, Person Production and Organizational Action», *Qual. Sociol.*, vol. (forthcoming), jun. 2015, doi: 10.1007/s11133-015-9302-7.

[17] R. Touray, «A Review of Records Management in Organisations», *Open Access Libr. J.*, vol. 8, n.o 12, Art. n.o 12, dic. 2021, doi: 10.4236/oalib.1108107.

[18] C. E. Fuentes Bolaños, «Redacción y presentación de informes», *Rev. Cienc. Adm. Financ. Segur. Soc.*, vol. 11, n.o 2, pp. 75-85, ene. 2003.

[19] «The function of documents - ScienceDirect». Accedido: 1 de agosto de 2024. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/abs/pii/S0262885698000687

[20] C. Nawei y D. Blostein, «Blostein, D.: A survey of document image classification: problem statement, classifier architecture and performance evaluation. International Journal of Document Analysis and Recognition (IJDAR) 10(1), 1-16», *IJDAR*, vol. 10, pp. 1-16, jun. 2007, doi: 10.1007/s10032-006-0020-2.

[21] B. Jung, W.-J. Lee, y D. Weber, «Financial Reporting Quality and Labor Investment Efficiency», *Contemp. Account. Res.*, vol. 31, ago. 2013, doi: 10.1111/1911-3846.12053.

[22] B. Potvin, R. Villemaire, y T. Le, «A Position-Based Method for the Extraction of Financial Information in PDF Documents», dic. 2016, pp. 9-16. doi: 10.1145/3015022.3015024.

[23] A. Godwin y I. Ramsay, «Financial Products and ShorttForm Disclosure Documents Challenges and Trends», *SSRN Electron. J.*, ene. 2014, doi: 10.2139/ssrn.2510189.

[24] J. Mleczko y L. Dulina, «Manufacturing Documentation for the High-Variety Products», *Manag. Prod. Eng. Rev.*, vol. 5, ene. 2014, doi: 10.2478/mper-2014-0027.

[25] A. K. Kalia, H. R. Motahari-Nezhad, C. Bartolini, y M. P. Singh, «Monitoring Commitments in People-Driven Service Engagements», en *2013 IEEE International Conference on Services Computing*, jun. 2013, pp. 160-167. doi: 10.1109/SCC.2013.62.

[26] S. Deshmukh y C. Lee, *Adapting Task-Oriented Dialogue Models for Email Conversations*. 2022.

[27] R. Rameshkumar, P. Bailey, A. Jha, y C. Quirk, «Assigning people to tasks identified in email: The EPA dataset for addressee tagging for detected task intent», en *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, W. Xu, A. Ritter, T. Baldwin, y A. Rahimi, Eds., Brussels, Belgium: Association for Computational Linguistics, nov. 2018, pp. 28-32. doi: 10.18653/v1/W18-6104.

[28] E. Alpaydin, *Introduction to Machine Learning, fourth edition*. MIT Press, 2020.

[29] S. Ganguly, «Machine Learning», en *Quantum Machine Learning: An Applied Approach: The Theory and Application of Quantum Machine Learning in Science and Industry*, S. Ganguly, Ed., Berkeley, CA: Apress, 2021, pp. 41-97. doi: 10.1007/978-1-4842-7098-1\_2.

[30] C. Janiesch, P. Zschech, y K. Heinrich, «Machine learning and deep learning», *Electron. Mark.*, vol. 31, n.o 3, pp. 685-695, sep. 2021, doi: 10.1007/s12525-021-00475-2.

[31] R. Pugliese, S. Regondi, y R. Marini, «Machine learning-based approach: global trends, research directions, and regulatory standpoints», *Data Sci. Manag.*, vol. 4, pp. 19-29, dic. 2021, doi: 10.1016/j.dsm.2021.12.002.

[32] P. C. Wong, «Unsupervised Machine Learning», en *Computational Psychometrics: New Methodologies for a New Generation of Digital Learning and Assessment*, Springer, Cham, 2021, pp. 173-193. doi: 10.1007/978-3-030-74394-9\_10.

[33] M. van Otterlo y M. Wiering, «Reinforcement Learning and Markov Decision Processes», en *Reinforcement Learning: State-of-the-Art*, M. Wiering y M. van Otterlo, Eds., Berlin, Heidelberg: Springer, 2012, pp. 3-42. doi: 10.1007/978-3-642-27645-3\_1.

[34] X. Chen, H. Xie, y X. Tao, «Vision, status, and research topics of Natural Language Processing», *Nat. Lang. Process. J.*, vol. 1, p. 100001, ene. 2022, doi: 10.1016/j.nlp.2022.100001.

[35] A. Rayhan, R. Kinzler, y R. Rayhan, «NATURAL LANGUAGE PROCESSING: TRANSFORMING HOW MACHINES UNDERSTAND HUMAN LANGUAGE», 26 de agosto de 2023. doi: 10.13140/RG.2.2.34900.99200.

[36] C. Huls, W. Claassen, y E. Bos, «Automatic Referent Resolution of Deictic and Anaphoric Expressions», vol. 21, jul. 2002.

[37] K. Liu, W. Hogan, y R. Crowley, «Natural Language Processing Methods and Systems for Biomedical Ontology Learning», *J. Biomed. Inform.*, vol. 44, pp. 163-79, feb. 2011, doi: 10.1016/j.jbi.2010.07.006.

[38] P. Turney y P. Pantel, «From Frequency to Meaning: Vector Space Models of Semantics», *J. Artif. Intell. Res.*, vol. 37, mar. 2010, doi: 10.1613/jair.2934.

[39] A. Ivankov, «Natural Language Processing: Advantages and Disadvantages», Profolus. Accedido: 30 de junio de 2024. [En línea]. Disponible en: https://www.profolus.com/topics/natural-language-processing-advantages-and-disadvantages/

[40] «What is Unified Modeling Language (UML)?» Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/

[41] «Qué es el lenguaje unificado de modelado (UML)», Lucidchart. Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml

[42] «IBM Rational Unified Modeling Language.» Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://www.kpms.ru/EN\_automation/EN\_Rational\_Rose.htm

[43] P. Turmero, «Introducción a Rational Rose», Monografias.com. Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://www.monografias.com/trabajos105/introduccion-rational-rose/introduccion-rational-rose

[44] «Visual Paradigm user’s guide». Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://www.visual-paradigm.com/support/documents/vpuserguide.jsp

[45] «Visual Paradigm». Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://www.oit.va.gov/Services/TRM/ToolPage.aspx?tid=10208

[46] «Visual Paradigm Product Overview». Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963\_visualparadi.html

[47] S. Rohn, «What Is a Desktop Application? +Challenges, Use Cases», The Whatfix Blog | Drive Digital Adoption. Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://whatfix.com/blog/desktop-application/

[48] «Aplicaciones Web Vs Aplicaciones de Escritorio - Mariano Gendra | Servicios de Desarrollo y Consultoria IT», Mariano Gendra. Accedido: 16 de julio de 2024. [En línea]. Disponible en: http://marianogendra.com.ar/Articulos/aplicaciones-web-vs-escritorio

[49] «What is Web Application (Web Apps) and its Benefits», Software Quality. Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app

[50] TheKnowledgeAcademy, «What is a Mobile App: Everything You Need to Know». Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://www.theknowledgeacademy.com/blog/what-is-a-mobile-app/

[51] «Aplicación móvil frente a aplicación web». Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://www.alegria.group/es/blog/mobile-app-vs-web-app

[52] «What is Python? Executive Summary», Python.org. Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://www.python.org/doc/essays/blurb/?external\_link=true

[53] «Introduction to Python». Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://www.w3schools.com/python/python\_intro.asp

[54] «Introduction to Java». Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://www.w3schools.com/java/java\_intro.asp

[55] «What is Java?—Beginner’s Guide to Java | Microsoft Azure». Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-java-programming-language

[56] «What is C++?», KO2 Recruitment. Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://www.ko2.co.uk/what-is-c-plus-plus/

[57] «What Is C++ (And How to Learn It)», Coursera. Accedido: 16 de julio de 2024. [En línea]. Disponible en: https://www.coursera.org/gb/articles/what-is-c-plus-plus

[58] «NLTK :: Natural Language Toolkit». Accedido: 17 de julio de 2024. [En línea]. Disponible en: https://www.nltk.org/

[59] «What is spaCy? | Domino Data Lab». Accedido: 17 de julio de 2024. [En línea]. Disponible en: https://domino.ai/data-science-dictionary/spacy

[60] «Models & Languages · spaCy Usage Documentation», Models & Languages. Accedido: 17 de julio de 2024. [En línea]. Disponible en: https://spacy.io/usage/models

[61] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, y D. McClosky, «The Stanford CoreNLP Natural Language Processing Toolkit», en *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 55-60. doi: 10.3115/v1/P14-5010.

[62] «Welcome | FreeLing Home Page». Accedido: 17 de julio de 2024. [En línea]. Disponible en: https://nlp.lsi.upc.edu/freeling/node/1

[63] «Comparison Between Angular vs React vs Vue | Angular Minds». Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://www.angularminds.com/blog/comparison-between-angular-vs-react-vs-vue

[64] «Qué es Vue JS y qué lo diferencia de otros frameworks | OpenWebinars», OpenWebinars.net. Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://openwebinars.net/blog/que-es-vue-js-y-que-lo-diferencia-de-otros-frameworks/

[65] @dmin, «Quels sont les meilleurs frameworks Vue.js à utiliser en 2022 ?», Mobiskill. Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://mobiskill.fr/blog/conseils-emploi-tech/quels-sont-les-meilleurs-frameworks-vue-js-a-utiliser-en-2022/

[66] «Why Is React a Library and Next.js a Framework?», Built In. Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://builtin.com/software-engineering-perspectives/react-framework

[67] «What is React.js? Uses, Examples, & More». Accedido: 15 de julio de 2024. [En línea]. Disponible en: https://blog.hubspot.com/website/react-js

# Glosario de términos

Aplicación Web: Son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador.

CASE: Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora). Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de la misma en términos de tiempo y de dinero.

Framework: En informática es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

HTML: (HyperText Markup Language o Lenguaje de Etiquetas de Hipertexto): Es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

JavaScript: Es un lenguaje de programación interpretado. Se define como orientado a objetos.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

MVC: Acrónimo del patrón Modelo, Vista, Controlador. Es un patrón clásico de diseño Web.

MVVM: Acrónimo del patrón Modelo, Vista, Vista, Modelo. Es un patrón de diseño.

Software: Programa de aplicación o sistema operativo que una computadora puede ejecutar, el termino software es muy amplio y se puede referir tanto a un solo programa como a varios programas, también se puede referir a aplicaciones que en realidad puede que las integre más de un solo programa.

UML: Unified Modeling Language. Lenguaje que permite la modelación de sistemas con tecnologías orientadas a objetos.

Web: sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio sí se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.

XML: Acrónimo inglés de Xtensible Markup Language (Lenguaje extensible de marcas). Es una nueva tendencia y uno de sus propósitos es facilitar el establecimiento de diferencias entre la vista de un documento y su contenido.