



Evaluación de técnicas de clustering para agrupar Artefactos de Requerimientos: Escenarios
escritos en Lenguaje Natural

Tutor Técnico: Dr. Leandro Antonelli

Profesora: Dra. Marcela Rosalba Samela

Estudiante: Lovena Laurent

Trabajo Final de Carrera para Obtener el Título de Licenciatura en Gestión De Tecnología
Informática

Noviembre 2025

Resumen

La ingeniería de requerimientos es una etapa clave en el ciclo de vida del desarrollo de software. Es fundamental contar con un modelo de evaluación que ayude a las empresas de software a medir su capacidad para aplicar la ingeniería de requerimientos. Dificultades en las tareas del proceso, como planes poco claros, que se registran de manera incompleta, pueden resultar al fracaso del proyecto, que se entreguen productos de calidad deficiente y que se superen los presupuestos establecidos. En dominios complejos, la utilización de escenarios para capturar requerimientos ha despertado un interés creciente entre los profesionales de la ingeniería de requerimientos. La cantidad de escenarios posibles puede ser inmenso, lo que requiere técnicas avanzadas. La manera de abordar esta complejidad es agrupando los escenarios bajo algún criterio, aplicando técnicas de clustering. La evaluación de técnicas de clustering para agrupar artefactos de requerimientos en lenguaje natural es un aspecto fundamental en la ingeniería de software y el procesamiento del lenguaje. Existen varias técnicas y estrategias para llevar a cabo esta actividad, no obstante, seleccionar la técnica de agrupación correcta puede ser complicada, ya que no hay una solución universal. Por ese motivo, la intención de este trabajo final es evaluar técnicas de clustering para agrupar escenarios con el uso de Modelo de Lenguaje de gran escala que tiene la capacidad de optimizar las técnicas tradicionales de agrupación de texto al enfocarse en la importancia de los conceptos, haciendo que sean más eficientes en la organización de textos.

Palabras clave: agrupamiento, escenarios, modelo de lenguaje de gran escala, procesamiento lenguaje natural, requerimientos

Abstract

Requirements engineering is a key stage in the software development lifecycle. It is essential to have an assessment model that helps software companies measure their ability to apply requirements in engineering. Difficulties in process tasks, such as unclear plans that are incompletely recorded, can lead to project failure, poor-quality products being delivered, and budget overruns. In complex domains, the use of scenarios to capture requirements has sparked growing interest among requirements engineering professionals. The number of possible scenarios can be immense, requiring advanced techniques. The way to address this complexity is by grouping scenarios under a certain criteria, applying clustering techniques. The evaluation of clustering techniques for grouping natural language requirement artifacts is a fundamental aspect of software engineering and language processing. There are several techniques and strategies for carrying out this activity; however, selecting the correct clustering technique can be complicated, as there is no universal solution. For this reason, the purpose of this final project is to evaluate clustering techniques for grouping scenarios using a Large Language Model (LLM), which can optimize traditional text clustering techniques by focusing on the importance of concepts, making them more efficient in organizing texts.

Keywords: clustering, large language model, natural language processing, requirements, scenarios

Dedicatoria

Dedico este trabajo a mi familia, por su apoyo incondicional, por ser mi mayor fuente de fortaleza.

A mis amigos y colegas, por acompañarme en este camino.

Y a todas las personas que, de una u otra forma, han sido parte de esta trayectoria gracias a todos.

Reconocimiento

A mi querida familia por su constante apoyo a lo largo de este proceso.

Al equipo de investigación de CAETI especialmente al Dr Antonelli Leandro y su equipo, por brindarme su conocimiento, tiempo y experiencia guiándome para llevar adelante este trabajo.

A mis colegas, quienes me han acompañado a lo largo de toda la carrera, en cada etapa y de diversas maneras. A todos ellos, mi más sincero agradecimiento por su apoyo y generosidad.

Índice General

Resumen.....	2
Abstract.....	3
Dedicatoria	4
Reconocimiento	5
Capítulo 1 - Introducción	9
1.1 Identificación del problema	9
1.2 Planteamiento del tema	10
1.3 Pregunta de Investigación	11
1.4 Hipótesis.....	11
1.5 Objetivo general.....	11
1.6 Objetivos específicos	11
1.7 Justificación del tema	12
1.8 Metodología.....	13
1.9 Fundamentación.....	13
Capítulo 2 - Estado del arte	14
Capítulo 3 - Marco teórico	24
3.1 Tipos de algoritmos de clustering.....	24
3.2 Escenarios usado para el análisis.....	25
3.2.1 <i>Clustering jerárquico</i>	43
3.2.2 <i>Clustering basado en particiones</i>	48
3.2.3 <i>Clustering basado en centroides</i>	51
3.2.4 <i>Clustering basado en densidad</i>	53
Capítulo 4 - Rendimientos de las técnicas	55
4.1 Rendimiento clustering jerárquico.....	56
4.2 Rendimiento clustering basado en particiones.....	57
4.3 Rendimiento clustering basado en centroides	59
4.4 Rendimiento Clustering basado en densidad	61
Capítulo 5 - Combinaciones de las técnicas.....	63
5.1 Combinación de jerárquico y particiones.....	63
5.2 Combinación de jerárquico y centroides	66
Capítulo 6 - Análisis.....	68

6.1 Uso con artefactos Escenarios	68
6.2 Evaluación de modelos	69
Capítulo 7 - Resultados.....	70
7.1 Diseño con uso de LLM	70
Capítulo 8 - Conclusiones.....	71
8.1 Líneas Futuras de Investigación	71
Acrónimos.....	72
Referencias	73

Índice de tablas

Tabla 1 Escenario 1 – Solicitud de estudio	25
Tabla 2 Escenario 2 - Turno para Extracción	27
Tabla 3 Escenario 3 en el Domicio de medicina – Envío de muestra	29
Tabla 4 Escenario 4 – Registrar nuevo Médico	30
Tabla 5 Escenario 5 - Registrar Nuevo Laboratorio	32
Tabla 6 Escenario 6 – Visualizar la lista de Usuarios	33
Tabla 7 Escenario 7 - Registrar un nuevo paciente en el sistema.....	34
Tabla 8 Escenario 8 - Generar presupuesto para un paciente	36
Tabla 9 Escenario 9 - Confirmar pago de presupuesto para estudio de un paciente	38
Tabla 10 Escenario 10 - Cargar los resultados del estudio del paciente	39
Tabla 11 Escenario 11 – Cierre de jornada laboral	40
Tabla 12 Escenario 12 - Confirmar asistencia del paciente al turno	42

Índice de Gráficos

Figura 1	Algoritmo Aglomerativo en base a los títulos de los escenarios presentados	45
Figura 2	Clustering Aglomerativo de mayores separaciones	46
Figura 3	Clustering Clarans en base a los escenarios en dominio clínico	49
Figura 4	Agrupamiento centroe - Kmeans.....	51
Figura 5	Agrupamiento en densidad - DBSCAN.....	53
Figura 6	Grafico de rendimiento de clustering aglomerativo	56
Figura 7	Métricas de rendimiento de clustering Clarans	58
Figura 8	Métricas de rendimiento de clustering Kmeans	60
Figura 9	Distribución cluster Scatter Plot DBSCAN.....	61
Figura 10	Comparación de las técnicas Clarans y Aglomerativo.....	63
Figura 11	Resultado de Clustering combinado.....	64
Figura 12	Comparación de las técnicas Kmeans y Aglomerativo	67

Capítulo 1 - Introducción

La ingeniería de requisitos constituye una etapa esencial en la creación de software, dado que establece el fundamento para el diseño, la ejecución y el mantenimiento de sistemas que atiendan las demandas del usuario. No obstante, en proyectos de gran escala, la recopilación, el análisis y la validación de requisitos pueden volverse complejas, inciertas y susceptibles de errores. En este contexto, el empleo de escenarios se ha establecido como una metodología eficaz para diseñar situaciones de uso reales, facilitando la identificación y validación de requisitos. El uso de escenarios para capturar requerimientos ha despertado un interés creciente entre los profesionales de la ingeniería de requerimientos. Dificultades en la tarea de organización y recolección propensa a errores y redundancia cuando la cantidad de escenarios posibles son inmensos, lo que requiere técnicas avanzadas. Es particularmente relevante en proyectos donde los escenarios provienen de múltiples fuentes como usuarios, interesados, documentación previa, lo que puede ocasionar contradicciones.

La implementación de algoritmos de clustering en este contexto no solo optimiza la eficiencia del proceso de análisis de requisitos, sino que también puede enriquecer la calidad del producto final al promover una comprensión más precisa de las necesidades del sistema. La presente investigación se enfoca en técnica de Clustering para identificar patrones similares con el propósito de mejora en la arquitectura del sistema.

1.1 Identificación del problema

Con la creciente cantidad de escenarios en ámbitos complejos, se dificulta su clasificación estructurada. A pesar de la existencia de métodos tradicionales para estructurar y depurar

requerimientos, estos suelen ser manuales, consumir mucho tiempo y depender del juicio personal de los analistas. En este contexto, se hace evidente la necesidad de respaldar este proceso con herramientas automáticas que faciliten la identificación de patrones, similitudes o agrupaciones en los escenarios descritos. Si bien existen diversas técnicas de agrupamiento, no es claramente definido si se pueden aplicar para organizar dominios y subdominios en distintos escenarios. Por eso surge la pregunta de si es viable detectar eficientemente patrones mediante escenarios aplicando técnicas de clustering, agrupamiento no supervisado. Se ha realizado una limitada investigación sistemática sobre la aplicación de algoritmos de agrupamiento en el ámbito específico de la ingeniería de requerimientos fundamentada en escenarios. Además, no está claro qué elementos de los escenarios deben considerarse para lograr un agrupamiento eficaz, ni cómo interpretar los resultados del agrupamiento de manera que sean útiles para los ingenieros de software.

1.2 Planteamiento del tema

Los escenarios ilustran circunstancias específicas de aplicación desde el punto de vista de los usuarios o las partes interesadas implicadas en el sistema. Sin embargo, en proyectos complejas que involucran a diversos interesados, la creación de escenarios puede aumentar de forma considerable, lo que resulta en desafíos vinculados a la redundancia, inconvenientes y dificultades para clasificar y examinar los datos de manera eficaz. En la actualidad, gran parte de estas labores se lleva a cabo de manera manual, lo que da lugar a procesos lentos, susceptibles a errores y dependientes del juicio subjetivo del analista.

La organización de dominios y subdominios dentro de sistemas complejos en entornos tecnológicos plantea un desafío considerable. La incapacidad para clasificarlos y organizarlos de manera efectiva puede limitar el diseño y la estructura del software. Una posible solución a esta problemática es la agrupación lógica de patrones en función de determinados criterios. Las técnicas de clustering provenientes del aprendizaje automático, presentan un enfoque prometedor para abordar estos desafíos, permitiendo agrupar escenarios similares y analizar grandes volúmenes de información de manera estructurada, la utilización de escenarios como representaciones contextuales podría facilitar la identificación de comportamientos recurrentes.

Esta tesis se dedicará a analizar diversos escenarios con la finalidad de identificar patrones relevantes que promuevan la estructura y organización de dominios y subdominios, contribuyendo, de este modo, a una mejora en la arquitectura del sistema.

1.3 Pregunta de Investigación

¿Es posible aplicar técnicas de Clustering para identificar patrones que organicen dominios y subdominios a partir de escenarios?

1.4 Hipótesis

En sistemas complejos en el entorno tecnológico, la cantidad de dominios y subdominios posibles puede ser inmenso, detectar patrones mediante el análisis de escenarios permite una organización más sistemática y consistente de estos componentes.

1.5 Objetivo general

Evaluar diferentes técnicas de clustering para facilitar la agrupación de elementos emergentes a partir de escenarios, con el propósito de detectar patrones que permiten organizar dominios y subdominios en ingeniería de requerimientos.

1.6 Objetivos específicos

- Identificar las técnicas de clustering existentes aplicables al análisis estructurado de escenarios.

- Comparar el rendimiento de distintos algoritmos de agrupamiento mediante su combinación en contextos de análisis de escenarios.
- Diseñar prompts adecuados para la exploración mediante el uso de modelos de lenguaje de gran escala (LLM).
- Analizar los resultados obtenidos por los algoritmos en la detección de patrones relevantes para la organización de dominios y subdominios.

1.7 Justificación del tema

La revisión manual de escenarios, en particular cuando son extensas y diversas, presenta diversas limitaciones: requieren una considerable inversión de tiempo, sujeta a errores humanos y depende en gran medida de la experiencia del analista. Asimismo, cuando hay un número elevado de situaciones, a menudo se manifiestan redundancias, y contradicciones que comprometen la claridad y coherencia de los requisitos generados.

En este marco, las metodologías de agrupamiento presentan una solución innovadora y automatizable para abordar estas dificultades. Dichas metodologías permiten identificar conjuntos de situaciones similares mediante el análisis de características compartidas, lo que simplifica la organización, el análisis y la validación de los requisitos. A pesar de su potencial, el agrupamiento ha recibido escasa atención dentro del marco formal de la ingeniería de requisitos, en particular en su aplicación práctica con situaciones narrativas.

La correcta organización de dominios y subdominios es fundamental en disciplinas como la ingeniería de requisitos e ingeniería de software, una clasificación deficiente puede afectar el análisis, diseño y desarrollo del software. El uso de los escenarios como representaciones contextuales podría facilitar la identificación de comportamientos recurrentes en la ingeniería de requisitos, tiene el potencial de identificar e estructurar soluciones recurrentes a patrones similares dentro de un dominio específico.

Esta investigación es relevante porque propone una metodología alternativa, basada en escenarios, que podría ser aplicables a varias disciplinas.

1.8 Metodología

La investigación ha tenido un enfoque cuantitativo y experimental, usando lenguaje gran escala para preparar los datos, luego se aplicarán distintos algoritmos de clustering. Se comparará los resultados de las distintas técnicas. Los resultados se evalúan con herramientas de métricas para determinar rendimientos.

Herramienta y Software

Los datasets se han generado a través de escenarios de la CAETI, los algoritmos se han generado con ChatGPT con la generación de código Python, el análisis ha implementado en Google Colab utilizando las bibliotecas Scikit-learn, Pandas y Matplotlib para el procesamiento, clustering y visualización de datos. Luego los resultados se han evaluados con métricas como Silhouette Coefficient, Davies–Bouldin Index, Calinski–Harabasz Index para medir el rendimiento de los algoritmos.

1.9 Fundamentación

En la actualidad, los sistemas están aumentando en complejidad y requieren métodos más eficientes para organizar y analizar requisitos. Uno de los mayores retos durante esta fase del desarrollo es presentar el entendimiento del dominio de forma clara, especialmente cuando los requisitos se describen en escenarios usando lenguaje común.

En este sentido, las técnicas de procesamiento de lenguaje natural y agrupamiento facilitan la identificación de relaciones semánticas entre los distintos escenarios.

Investigar la aplicación de clustering en este ámbito podría tener un impacto significativo en la organización de la información en las fases iniciales del desarrollo, permitir la detección de redundancias y establecer bases para arquitecturas más coherentes.

Capítulo 2 - Estado del arte

Este trabajo está basado en la ingeniería de requerimientos con los artefactos de escenarios en procesamiento en lenguaje natural.

La ingeniería de requerimientos representa una fase crucial en el ciclo de vida del desarrollo de un sistema, es el proceso de examinar, documentar y gestionar las demandas de las partes interesadas en un proyecto de desarrollo. Este procedimiento transforma las necesidades y conceptos del cliente en especificaciones concretas. Su meta incluye una comunicación eficiente de todos los involucrados y garantizar que el producto final cumpla con las demandas y necesidades definidas. (Mazak-Huemer et al., 2024)

Existen múltiples enfoques de ingeniería de requerimientos de los cuales podemos citar de diferentes artículos.

Según Celi et al. (2023), los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información. Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama ingeniería de requerimientos (IR). (p.47)

También se puede definir la ingeniería de requerimientos según Glinz M (2020) como: Un método estructurado y riguroso para definir y manejar los requisitos, cuyo propósito es asegurar una comprensión completa y precisa de dichos requerimientos, minimizando así el riesgo de desarrollar un sistema que no satisfaga adecuadamente sus demandas. Este proceso contribuye a mejorar la calidad del producto final y a garantizar que los resultados estén alineados con los objetivos establecidos desde el inicio del proyecto. (p.17)

Es crucial realizar esta actividad en la etapa temprana del proyecto dada que permite establecer una definición precisa y detallada de los requisitos, proporciona un fundamento para el diseño y ejecución del sistema, lo que favorece el éxito del proyecto a largo plazo. Abordar la ingeniería de requerimientos con seriedad desde la fase inicial es fundamental para establecer las

bases de un desarrollo de sistema exitoso, eficaz y enfocado en el usuario. Esta metodología ayuda a diseñar productos que sean más útiles, eficientes, sostenibles y que responden adecuadamente a las necesidades de los usuarios. Implementar este proceso desde el comienzo no solo eleva la calidad del producto, sino que también contribuye a la reducción de costos, previene trabajos duplicados y alinea el desarrollo con las metas. Esto evita la creación de funcionalidades innecesarias y favorece la priorización de las que aportan un mayor valor. Además, permite la asignación efectiva de las tareas. (Pérez G. et al., 2024)

Tal como menciona Mazak-Huemer et al. (2024), Las actividades vinculadas a la ingeniería de requisitos generalmente se llevan a cabo en las fases tempranas del ciclo de vida del desarrollo de software. Esto comienza con la evaluación, la recolección y la formalización de los objetivos deseados, además de la detección de las dificultades que podrían interferir la adecuada ejecución del sistema. (p.2)

En este contexto, el Procesamiento del Lenguaje Natural (PLN) se ha convertido como un recurso esencial para mejorar y automatizar diversas actividades relacionadas con la ingeniería de requisitos. Dada que el lenguaje natural es el método más común a través del cual los clientes comunican sus requisitos a los desarrolladores de software ya sea entrevistas o documentos. Aunque esta forma de interacción es tanto accesible como versátil, también enfrenta diversos retos que podrían afectar la precisión y la eficacia en el proceso de recopilación de requisitos. Porque están escritos en lenguaje natural propensa a generar mal entendimientos por uso de dominio distintos dependiendo del área, como las partes interesadas y el equipo de desarrollo tienden a tener forma distinta de expresarse. Además, el lenguaje natural carece de formalismo, lo que hace más complicado su análisis y la validación coherente de los requisitos. Los requisitos deben tener condiciones de aceptación bien definidas que eliminen cualquier duda y faciliten la verificación de cuándo se cumple un requisito. Una mala interpretación de los requisitos puede traer serias consecuencias en el proyecto tales como:

- Repetición de trabajo, desarrollan funciones que después tienen que rehacerse.
- Descontento del cliente, el sistema no satisface sus expectativas.

(Necula, S.-C et al., 2024 ; Pérez G. et al., 2024 ; Zhao L. et al., 2021).

Debido a la estrecha relación entre lenguaje natural y requerimientos se han desarrollado herramientas y métodos de procesamiento del lenguaje natural para el proceso de especificación de requisitos, gracias a las mejoras tecnológicas en PLN, actualmente existen soluciones que facilitan diversas tareas dentro de la ingeniería de requerimientos. (Zhao L. et al., 2021)

En este contexto se introduce NLP4RE, que Zhao L. et al. (2021), define como un área de investigación y desarrollo de soluciones que integren tecnologías de PLN incluyendo métodos, herramientas y recursos en el análisis de documentos de requisitos. Su propósito es proporcionar soporte sistemático a las tareas de procesamiento lingüístico que se llevan a cabo en las distintas etapas del ciclo de vida de la ingeniería de requisitos. (p.5)

La elicitación de requerimientos es una de las etapas más importantes en el ciclo de vida del desarrollo de software. En esta etapa, los analistas de requisitos necesitan recoger datos detallados de las partes interesadas como los clientes, usuarios finales, y expertos en el área de negocio para saber qué funcionalidades debe tener el sistema. Esta interacción se lleva a cabo, principalmente, usando lenguaje natural, ya sea de manera oral o escrita. En proyectos complicados, diversas partes interesadas pueden comunicar necesidades opuestas sin darse cuenta. Como el lenguaje común no sigue una estructura rígida, es sencillo que surjan incoherencias. Identificar estas incoherencias exige un análisis detallado de los textos y declaraciones. Se debe evitar los términos imprecisos que carecen de criterios objetivos. Si no se traducen en requisitos que se puedan medir y verificar, generan dificultades en la validación del sistema. (Bernanda et al., 2024; Lim et al., 2021)

Aunque la comunicación en lenguaje natural es la manera más sencilla de interactuar entre personas, también presenta un gran desafío cuando se emplea para elicitar requerimientos de sistemas. El desafío que genera el uso de lenguaje natural para capturar requisitos ocurre debido a que una misma palabra puede poseer varios significados, dependiendo del área en que se aplique. Por ello, resulta crucial complementar el lenguaje natural con métodos formales y herramientas de modelado que proporcionen una representación más exacta y verificable de las necesidades del usuario. Esta falta de semántica puede complicar la adecuada interpretación de los requisitos, dificulta la extracción de características relevantes y eso impacta en la calidad de las especificaciones (Bernanda et al., 2024, Zhao L. et al., 2021)

Tal como menciona Necula, S.-C et al. (2024), uno de los desafíos más importantes al usar el procesamiento del lenguaje natural (PLN) para la ingeniería de requisitos de software es gestionar la ambigüedad y complejidad inherentes al lenguaje natural. Las herramientas de PLN suelen tener dificultades para interpretar el contexto con precisión, lo que puede generar malentendidos en la obtención y documentación de requisitos. (p.19)

Para evitar ambigüedades y mal entendimientos, es fundamental modelar el dominio de forma estructura. El término dominio refiere al conjunto de normas y procedimientos que definen el área particular para la que se elabora un sistema. Se trata del contexto que establece el problema que se debe abordar y las demandas del usuario. El dominio es el fundamento desde el cual se crean los requisitos. Tener un conocimiento sólido de este permite construir sistemas que solucionen de verdad los problemas de los usuarios y se adapten a su situación. Pasar por alto el dominio puede resultar en soluciones que no se ajusten bien, sean caras y no sean bien recibidas. Así que, la ingeniería de requerimientos no se limita solo a recopilar lo que los usuarios necesitan, sino que implica comprender en profundidad por qué y dónde lo necesitan: eso es lo que se conoce como dominio. Conocer el área es crucial para recoger correctamente los requerimientos, que son vitales para conseguir un buen producto. Por esto, sin entender el área, es casi inviable crear una aplicación efectiva. Cada dominio tiene su propio vocabulario técnico o específica. Comprender esta terminología es fundamental para evitar confusiones durante la recolección de requisitos. El dominio es fundamental para entender bien el problema que se quiere solucionar con un sistema. Pasar por alto el dominio o tratarlo de manera superficial es una de las razones principales por las cuales los proyectos de software fracasan. La ingeniería de requerimientos actual requiere que los analistas no solo conozcan métodos para recopilar y documentar información, sino que también sean capaces de adaptarse de manera rápida y eficaz a diferentes dominios, entendiendo su lenguaje y necesidades. (Latif et al., 2018 ; Pérez G. et al., 2024)

Según Lim et al. (2021), tradicionalmente la recopilación de requisitos ha dependido de la comunicación directa con las partes interesadas, utilizando técnicas tradicionales como entrevistas, talleres, encuestas y grupos de discusión. Si bien este método, que es principalmente cualitativo, ha demostrado ser exitoso en múltiples contextos, también tiene limitaciones significativas: exige un considerable compromiso de tiempo, depende en gran medida de la disposición de los

participantes involucrados y no es de fácil escalabilidad en entornos organizativos que están distribuidos a nivel mundial. (p.1)

Con el avance de la digitalización, se desarrollaron herramientas que intentaban automatizar ciertos procesos asociados a requisitos estáticos como algoritmos para transformar modelos de negocio o diagramas UML en especificaciones, y técnicas de clasificación de requisitos ya definidos en categorías funcionales y no funcionales. No obstante, todo eso resulta insuficiente frente a la realidad actual, los sistemas modernos se despliegan en ecosistemas digitales cambiantes, donde las expectativas de los usuarios y las condiciones de uso evolucionan en cuestión de días. La dependencia exclusiva de documentos y reuniones presenciales limita la capacidad de respuesta y reduce la competitividad de las organizaciones. (Lim et al., 2021; Abdalazeim y Meziane 2021)

Este panorama plantea desafíos respecto a cómo interpretar, estructurar y aprovechar dicha información no estructurada. En el ámbito de la ingeniería de requisitos, se entiende por datos no estructurados aquella información relevante para la obtención y estudio de requisitos que no se presenta dentro de un formato formal establecidos. Este tipo de datos abarca documentos redactados en lenguaje natural como transcripciones de entrevistas o correos electrónicos que, a pesar de su falta de una estructura definida, poseen información valiosa sobre las necesidades, expectativas y limitaciones de los interesados. Para manejar estos datos, es necesario emplear técnicas específicas como el análisis cualitativo, la minería de texto o el procesamiento del lenguaje natural, con el objetivo de descubrir y sistematizar requisitos que sean útiles para el desarrollo de sistemas. Se utilizan diversas técnicas con el propósito de gestionar los requerimientos de un sistema. Estas técnicas se adaptan dependiendo del tipo de proyecto, la característica del dominio, el nivel de complejidad del sistema y los interesados. Algunas de estas incluyen entrevistas, brainstorming, modelos formales, análisis de tareas y creación de escenarios, entre otras. (Lim et al., 2021)

En este trabajo, se enfoca en la utilización del artefacto de escenarios como una técnica fundamental dentro del proceso. Este artefacto se define como la descripción minuciosa de situaciones específicas que ilustran cómo un sistema será empleado en un entorno real, lo que

permite comprender no solo las funcionalidades que debe poseer el sistema, sino también el porqué, cómo y bajo qué circunstancias. A través de los escenarios, se pueden identificar funcionalidades deseadas, posibles excepciones, comportamientos alternativos y limitaciones contextuales que no siempre se encuentran claramente expresadas en los requisitos convencionales escritos en un lenguaje natural. (Pérez G. et al., 2024)

El uso de escenarios constituye un enfoque centrado en el usuario que facilita la comprensión entre las partes interesadas, favorece la detección temprana de inconsistencias o de requisitos omitidos, y aporta valor en contextos tanto convencionales como ágiles. Su adaptabilidad y el bajo costo asociado a su implementación los convierten en una herramienta esencial dentro del estado actual de las mejores prácticas para la especificación de requisitos. (Pérez G. et al., 2024)

Asimismo, los escenarios tienen la capacidad de ser empleados en diversas etapas del desarrollo, comenzando desde la recopilación inicial de requisitos, continuando con el análisis, la validación, y finalizando con las pruebas del sistema, al ser transformados en casos de prueba basados en situaciones de uso reales. Su carácter específico y centrado en el usuario los hace particularmente eficaces para confirmar que el sistema satisface las necesidades reales del entorno operativo. (Pérez G. et al., 2024; Lim et al., 2021)

Al respecto Pérez G. et al. (2024) afirman lo siguiente: Los escenarios son herramientas útiles que permiten explicar cómo funciona un sistema a través de la narración de historias. Este enfoque es efectivo porque es una forma de incorporar detalles que son esenciales para una comprensión más clara y completa de su funcionamiento. Tanto los desarrolladores como los expertos pueden usarlos sin la necesidad de aprender formalismos complejos. Esto facilita la comunicación entre las partes interesadas y se pueden utilizar en diferentes etapas del desarrollo de software, para mejorar la comprensión del comportamiento esperado del sistema. (p.3)

Como que los datos se recuperan de forma no estructurada los escenarios se presentan como una herramienta particularmente valiosa para estructurar y organizar información no estructurada, permiten convertir la ambigüedad y la dispersión de la información no estructurada

en descripciones claras y precisas sobre el comportamiento que se anticipa del sistema, permite establecer de manera progresiva los requisitos basados en experiencias reales. No obstante, en ámbitos complejos cuando hay gran cantidad de información no estructurada, depender únicamente de métodos manuales para desarrollar escenarios resulta ineficaz y puede dar lugar a interpretaciones sesgadas. (Pérez G. et al., 2024; Necula, S.-C et al., 2024)

Para reducir ambigüedades y redundancias en contextos complejos caracterizados por una elevada cantidad de escenarios, las técnicas de agrupamiento emergen como herramientas fundamentales para la identificación y estructuración de patrones semánticos subyacentes. Según Lim et al. (2021) La técnica de agrupación tradicional se considera un método de aprendizaje automático no supervisado que busca descubrir la estructura fundamental en los datos. Esto se logra dividiendo un conjunto en clústers, de acuerdo con las relaciones de similitud y disimilitud.

Según Schleifer et al. (2025), El uso de algoritmos de agrupamiento es una práctica común para detectar patrones en datos que están representados a través de vectores como embeddings. Define Clustering como un método de inteligencia artificial no supervisado que utiliza una métrica de distancia para determinar cuán parecidos son los puntos y decidir si deben ser agrupados en el mismo conjunto. (p.4)

En este contexto las incrustaciones de palabras constituyen un método ampliamente utilizado para transformar los requisitos expresados en lenguaje natural en un formato comprensible para las máquinas, a través de su representación en vectores numéricos. (Schleifer et al., 2025, p.3)

Diversas investigaciones han explorado el uso de Clustering y procesamiento de lenguaje natural en el ámbito de la ingeniería de requerimientos. Como el trabajo de Ferrari Alessio et al. (2013) que aborda el uso de técnicas de agrupamiento en clústers como una estrategia para optimizar la disposición de los documentos de requisitos escritos en lenguaje natural. Define documento de requisitos como un documento que tiene buena organización cuando cada requisito está lógicamente conectado con los que están a su alrededor en la misma sección. Además, se aprecia la autonomía entre secciones cuando cada una está conceptualmente distinta de las demás.

En este contexto Ferrari Alessio et al. (2013) introduce el algoritmo de agrupamiento conocido como Sliding Head-Tail Component (S-HTC), el cual tiene como meta identificar las variaciones en las ideas que subyacen a los requisitos y exponer una estructura implícita dentro del texto fundamentada en esas variaciones. Esta estructura no evidente se genera al organizar los requisitos de acuerdo con su relación léxica, dividiendo las secciones según su independencia en el lenguaje. El algoritmo S-HTC reúne requisitos que están cercanos en el texto y que presentan semejanzas en su redacción. Este método intenta imitar la manera en la que un lector comprende un documento: si dos requisitos están situados uno al lado del otro y tienen similitudes lingüísticas, es probable que aborden un mismo tema. Concluyó que es eficaz para identificar áreas del documento con problemas estructurales, secciones que requieren reorganización y requisitos que han sido colocados en ubicaciones inadecuadas.

Bernanda et al. (2024), proponen aplicar Procesamiento de Lenguaje Natural en conjunto con algoritmos de clustering no supervisado K-means y MeanShift para mejorar la etapa de elicitación de requisitos con la meta de comprobar si estos algoritmos ayudan a organizar retroalimentación de usuarios y stakeholders de manera más eficiente, facilitando la identificación de necesidades en entornos universitarios. Introducen la combinación de NLP y clustering no supervisado en la etapa inicial de RE. Evalúan de forma comparativa K-means y MeanShift en un contexto real universitario. Luego proponen métricas cuantitativas y cualitativas para validar la utilidad de los agrupamientos. (p.4)

También el trabajo de Eyal-Salman et al. (2018), que sugiere un esquema general orientado a la agrupación automática de los requisitos funcionales del software, basándose en su significado semántico. Este esquema es adaptable a diversos algoritmos de agrupamiento y medidas de distancia. Además, propone un enfoque de agrupamiento jerárquico que facilita la organización semántica de los requisitos funcionales. Este enfoque comienza con una lista sencilla de enunciados de requisitos funcionales, similar a los que se suelen incluir en un documento de especificación de requisitos de software.

Por otro lado, Schleifer et al. (2025), proponen un sistema de tubería que utiliza inteligencia artificial y procesamiento de lenguaje natural para facilitar de manera automática la identificación de casos de uso a partir de especificaciones redactadas en lenguaje natural. La contribución principal se centra en dos elementos clave, el desarrollo de un grafo de requisitos que se apoya en un enfoque semi-supervisado, el cual integra tanto conocimientos previos, obtenidos a partir de mapeos incompletos elaborados manualmente, como representaciones de embeddings producidas por modelos de lenguaje como SBERT, y la implementación de técnicas de clustering de grafos con superposiciones para permitir que un requisito específico pueda ser relevante para múltiples casos de uso, lo cual refleja de manera real la complejidad inherente a los sistemas. Además, cada clúster se mejora con frases clave que ayudan a los ingenieros a interpretarlos de manera más efectiva. (p.8)

En base a esta investigación se determina que, si bien las técnicas de agrupamiento han mostrado ser útiles para organizar datos textuales, mejorar elicitación de requisitos e identificar temas que surgen, hay una escasez de enfoques que vinculen estos agrupamientos de manera directa con la creación de dominios y subdominios funcionales en el ámbito del desarrollo de software. Esta carencia presenta una oportunidad para progresar hacia técnicas que faciliten esta situación, partiendo del análisis de escenarios.

Este trabajo enfoca en aplicar técnicas como embeddings, índice Rand ajustado (ARI), el método de linkage Ward y los scatter plots que son esenciales para representar, evaluar y visualizar las estructuras ocultas en data sets complejos, como el trabajo de Zhao Jinyu et al. (2020) que crea un embedding no lineal que mejora la calidad de la agrupación luego aplica Ward linkage para la recuperación de estructuras jerárquicas, evidenciando mejoras que son cuantificables.

Se aplica el proceso de embedding que transforma los datos del escenario, en un espacio latente más fácil de manejar, conservando las relaciones semánticas y de similitud. Luego la aplicación linkage ward en clustering jerárquico Aglomerativo para mejorar la calidad del agrupamiento. Según Chacón et al. (2023) El Índice Ajustado de Rand (ARI) es frecuentemente empleado en el estudio de conglomerados para evaluar la medida en que dos particiones de datos coinciden. (p.6) La evaluación externa del clustering resulta clave, el ARI mide el grado de

coincidencia entre una partición obtenida y una partición de referencia, ajustando por lo que podría esperarse al azar. Compara la similitud entre dos particiones considerando todas las combinaciones posibles de pares de elementos.

Finalmente se genera la visualización de los clusters por medio de scatter plots que permite visualizar la distribución de los puntos en el espacio y si hay agrupaciones que ocurren de manera natural. Tras implementar un algoritmo de Clustering, se utiliza para mostrar los datos y se colorean los puntos de acuerdo con el grupo asignado. Cada color representa un grupo diferente lo que permite evaluar visualmente si los clusters están bien separados o si se superponen, se visualiza los grupos formados por un algoritmo y entender la estructura de los datos en el espacio de características reducido a 2D para poder graficarse. En este contexto, Jana y Barbora (2023), explican que cada punto del gráfico simboliza un conjunto de valores asociados a un caso particular. La posición del punto en los ejes horizontal y vertical permite visualizar cómo se relacionan dos variables numéricas entre sí. (p. 404)

Esta investigación aporta las siguientes contribuciones:

- Un enfoque de clustering para agrupar escenarios basado en su semántica.
- Un marco ampliable para incluir diferentes algoritmos de agrupamiento.
- Una investigación destinada a evaluar el agrupamiento empleando escenarios de acceso libre.
- Primero se clasifica información relevante mediante aprendizaje no supervisado, luego se agrupa en clústers temáticos y finalmente se visualiza para su interpretación.

Capítulo 3 - Marco teórico

3.1 Tipos de algoritmos de clustering

En la ingeniería de requerimientos, los artefactos de escenarios actúan como descripciones que ilustran la manera en que los usuarios se relacionan con el sistema en contextos concretos, permiten documentar el comportamiento esperado del sistema en un evento específico. (Mazak-Huemer et al., 2024; Pérez et al., 2024)

En esta sección proponemos estudiar distintos métodos de algoritmos de agrupamiento, y elegir la más eficiente para aplicar en artefactos de escenarios. En el estudio de escenarios, la combinación de diferentes algoritmos para mejorar la robustez en los resultados, sobre todo cuando se trabaja con datos variados o de múltiples dimensiones. Esta unión se ha llevado a cabo usando métodos de agrupamiento en conjunto y al comparar resultados para obtener patrones que se repiten.

Existen numerosos algoritmos de clustering, dado que hay diversas maneras de definir un clúster. Los distintos métodos funcionan en función del caso y volumen de los datos que se estén analizando. En esta sección se ha examinado cuatro métodos de agrupamiento. Se explica sus principales características, los algoritmos más utilizados en cada uno de ellos.

- Jerárquico como AGLOMERATIVE CLUSTERING
- Particiones como CLARANS
- Centroides como KMEANS
- Densidad como DBSCAN

3.2 Escenarios considerados en el análisis

En este trabajo enfocamos en 12 escenarios en el dominio de la medicina donde cada uno describe un escenario específico relacionado con el sistema, desde solicitud de un estudio, registrar un paciente, registrar nuevo médico hasta enviar muestra de laboratorio.

Tabla 1

Escenario 1 – Solicitud de estudio

Atributos	Descripción
Título	Solicitar un estudio para un paciente.
Objetivo	Realizar la solicitud de un estudio para diagnosticar la posible presencia de una enfermedad en un paciente.
Contexto	Un paciente previamente registrado presenta síntomas que sugieren la posibilidad de que sea una de las cinco enfermedades raras o quiere controlarse porque presenta antecedentes familiares, y el médico necesita realizar un estudio de ADN para confirmar o descartar la sospecha.
Actores	Médico
Recursos	• Información del paciente ya registrado en el sistema. • Síntomas que presenta el paciente y patología por la que sospecha. • API de validación de síntomas cardinales.

Atributos	Descripción
Episodios	• API obtener Genes por patología
	1. El médico visualiza una lista con sus pacientes, y elige al paciente al cual se le quiere solicitar un estudio.
	2. El médico selecciona la opción de solicitar un estudio para el paciente.
	3. El médico indica si se trata de una "sospecha puntual" o una "sospecha familiar" (en caso de tener antecedentes familiares).
	4. El médico selecciona la patología a evaluar
	5. El médico ingresa o selecciona de una lista los síntomas que presenta el paciente. (En caso de solicitar sospecha puntual)
	6. El médico, en caso de requerir, ingresa genes de una lista de genes adicionales a analizar.
	7. El médico, en caso de requerir, solicita analizar hallazgos secundarios.
	8. El médico solicita el estudio
	9. (En caso de sospecha puntual) El sistema solicita a la API verificar que al menos uno de los síntomas cardinales esté presente para la patología solicitada.

Atributos	Descripción
	10. (En caso de respuesta afirmativa de la API) El sistema solicita a la API los Genes de la patología a evaluar.
	11. El sistema registra el nuevo estudio con el estado “Solicitado” y notifica al paciente sobre la solicitud.

Nota. Elaboración propia: La tabla 1 describe un escenario específico en el dominio de medicina en el cual un médico desea solicitar un turno para uno de sus pacientes ya registrado en el sistema. Se elige este dominio porque permite agrupar estudios por tipo, patologías o síntomas.

Tabla 2

Escenario 2 - Turno para Extracción

Atributos	Descripción
Título	Solicitar un turno para la extracción de muestra.
Objetivo	Permitir que el paciente seleccione el lugar, día y hora para la extracción de su muestra.
Contexto	Tras aprobar el presupuesto, el paciente agenda un turno en uno de los centros de extracción disponibles y fecha libre, para tomar la muestra necesaria para el estudio.

Atributos	Descripción
Actores	Paciente
Recursos	Listado de Laboratorios disponibles. Turnos disponibles del Laboratorio seleccionado
Episodios	1. El paciente selecciona Solicitar turno sobre un Estudio previamente generado. 2. Elige la zona. 3. El paciente elige el centro. 4. El paciente elige el día. 5. El sistema muestra los horarios disponibles para sacar turno. 6. El paciente elige el horario. 7. El paciente carga el Consentimiento de Estudio, con la opción de descargar un modelo. 8. El sistema verifica el turno y notifica por correo electrónico la realización del turno.

Nota. Elaboración propia: La tabla 2 describe un escenario específico en el dominio de medicina en el cual un usuario con acceso a un laboratorio específico se realiza la extracción de un paciente. Se elige este dominio porque permite agrupar turnos por horarios, zonas y frecuencia de uso.

Tabla 3*Escenario 3 en el Domicio de medicina – Envío de muestra*

Atributos	Descripción
Título	Enviar muestras al exterior para análisis genético.
Objetivo	Realizar el envío de muestras al exterior.
Contexto	El laboratorio central ha acumulado el mínimo de 100 muestras. El sistema debe realizar el envío de las muestras al exterior para su análisis genético.
Actores	Laboratorio de extracción
Recursos	Listado de turnos.
Episodios	1. El Laboratorio selecciona “Turnos” 2. El sistema muestra una lista de los turnos ordenados de manera ascendente. 3. El Paciente se presenta y el laboratorio realiza la extracción 4. El laboratorio realiza la extracción a un paciente y apreta en “Confirmar Turno” del paciente al que realizó la extracción 5. El sistema actualiza el estado del turno a “ACEPTADO”, cambia el estado del estudio asociado a “ESPERANDO

Atributos	Descripción
	ENVIO A CENTRAL”, en caso de ser la primera extracción del día, el sistema genera un pedido para el transportista y notifica por mail al usuario el éxito de la extracción.

Nota. Elaboración propia: La tabla 3 describe un escenario específico en el dominio de medicina en el cual un laboratorio desea enviar las muestras al exterior para análisis. Se elige este dominio porque permite clasificar grandes volúmenes de muestras en grupos homogéneos por tipo, origen o estado, también permite agrupar sample sets por volumen, tiempos de envío o frecuencia de generación.

Tabla 4

Escenario 4 – Registrar nuevo Médico

Atributos	Descripción
Título	Registrar Médico
Objetivo	Permitir que el Owner registre un nuevo médico.

Atributos	Descripción
Contexto	Un nuevo usuario médico necesita ser registrado en el sistema para que pueda acceder y cumplir con sus funciones.
Actores	Owner
Recursos	Información del Médico: nombre, email, dni, teléfono, fecha de nacimiento.
Episodios	1. El Owner accede al sistema y selecciona la opción "Registrar Médico". 2. El sistema despliega un formulario con Nombre, Apellido, email, dni, teléfono, fecha de nacimiento. 3. El Owner completa cada campo con los datos del Médico. 4. Una vez completados los campos, el Owner hace clic en "Registrar Médico" para guardar al usuario en el sistema. 5. El sistema valida los datos ingresados, si todo es correcto, registra al nuevo usuario y le envía la contraseña por mail, asignándole una cuenta activa.

Nota. Elaboración propia: La tabla 4 describe un escenario específico en el dominio de medicina en el cual un Owner desea registrar un nuevo médico.

Tabla 5*Escenario 5 - Registrar Nuevo Laboratorio*

Atributos	Descripción
Título	Registrar Laboratorio
Objetivo	Permitir que el Owner registre un Laboratorio en el sistema.
Contexto	Se quiere ingresar un nuevo Usuario Laboratorio donde se pueda realizar las extracciones a los pacientes.
Actores	Owner
Recursos	Información del laboratorio: nombre, email, hora de apertura, hora de cierre, días de atención, teléfono y dirección
Episodios	1. El Owner accede al sistema y selecciona la opción "Registrar Laboratorio".

Atributos	Descripción
	2. El sistema despliega un formulario con Nombre, email, hora de apertura, Hora de cierre, días de atención, Teléfono, Dirección (con número o marcando en el mapa) 3. El Owner completa cada campo con los datos del Laboratorio. 4. Una vez completados los campos, el Owner hace clic en "Registrar Laboratorio" para guardar al usuario en el sistema. 5. El sistema valida los datos ingresados, si todo es correcto, registra al nuevo usuario y le envía la contraseña por mail, asignándole una cuenta activa.

Nota. Elaboración propia: La tabla 5 describe un escenario específico en el cual un Owner registra un nuevo laboratorio para extracción.

Tabla 6

Escenario 6 – Visualizar la lista de Usuarios

Atributos	Descripción
Título	Listar usuarios
Objetivo	Permitir que el Owner visualice la lista de usuarios (médico, transportista, administrador de laboratorio).

Atributos	Descripción
Contexto	El Owner necesita visualizar a los usuarios para su revisión o gestión en el sistema.
Actores	Owner
Recursos	Listado de Usuarios
Episodios	1. El Owner accede al sistema y elige la opción de "Ver Usuarios". 2. El sistema muestra un listado general de usuarios con sus datos principales. 3. El Owner revisa la lista de usuarios, teniendo la opción adicional para deshabilitar. 4. El sistema registra el nuevo estudio con el estado "Solicitado" y notifica al paciente sobre la solicitud.

Nota. Elaboración propia: La tabla 6 describe un escenario específico en el dominio de medicina en el cual un Owner desea visualizar los usuarios registrados del sistema.

Tabla 7

Escenario 7 - Registrar un nuevo paciente en el sistema

Atributos	Descripción
Título	Registrar Paciente
Objetivo	Ingresa en el sistema los datos de un paciente para registrarlo en el sistema.

Atributos	Descripción
Contexto	El médico derivante recibe a un paciente que presenta síntomas relacionados con una posible enfermedad rara, y debe registrarlo en el sistema para que se inicie el proceso de análisis y diagnóstico.
Actores	Médico derivante
Recursos	Información del paciente: nombre, apellido, correo electrónico, DNI, teléfono, fecha de nacimiento, antecedentes familiares y pdf de la historia clínica.
Episodios	1. El médico accede al sistema y selecciona la opción para registrar un nuevo paciente. 2. El médico introduce los datos personales del paciente: nombre, apellido, correo electrónico, DNI, fecha de nacimiento. 3. El médico completa los antecedentes familiares, seleccionando la enfermedad rara y el pariente que la padece/padeció. 4. El médico sube un resumen de la historia clínica relevante. 5. El médico guarda el registro del paciente en el sistema, finalizando así el proceso de alta.

Atributos	Descripción
	6. El sistema notifica por mail al Paciente con la contraseña.

Nota. Elaboración propia: La tabla 7 describe un escenario específico en el dominio de medicina en el cual un médico desea dar de alta un nuevo paciente en el sistema.

Tabla 8

Escenario 8 - Generar presupuesto para un paciente

Atributos	Descripción
Título	Generar presupuesto para un estudio solicitado.
Objetivo	Crear y registrar un presupuesto para un estudio genético solicitado por el médico para que el paciente pueda realizar el pago.
Contexto	Un médico ha solicitado un estudio genético para un paciente. El administrador de laboratorio debe generar un presupuesto en base a la cantidad de genes a analizar y si se solicitaron hallazgos secundarios.
Actores	Administrador de laboratorio
Recursos	Listado de estudios pendientes de presupuesto.

Atributos	Descripción
Episodios	1. El administrador de laboratorio accede al sistema y selecciona “Presupuestos Solicitados” 2. El sistema muestra el listado de estudios en estado “SOLICITADO” 3. El Administrador selecciona uno y se le muestra el detalle del estudio solicitando el precio base y ,en caso de requerir, el precio por gen adicional y hallazgos secundarios. 4. El administrador completa los precios requeridos. Además de colocar fecha de Vencimiento. 5. El administrador hace clic en "Generar presupuesto" para confirmar el cálculo. 6. El sistema registra el presupuesto, cambia el estado del estudio a “PRESUPUESTADO” y envía una notificación al paciente, indicando que el presupuesto está disponible para revisión y pago.

Nota. Elaboración propia: La tabla 8 describe un escenario específico en el dominio de medicina en el cual el administrador desea generar presupuesto para un cliente que debe realizar estudio de laboratorio.

Tabla 9*Escenario 9 - Confirmar pago de presupuesto para estudio de un paciente*

Atributos	Descripción
Título	Confirmar pago de presupuesto
Objetivo	Registrar el pago de un presupuesto generado para el estudio de un paciente.
Contexto	Un presupuesto de un estudio enviado a un paciente para su pago. El administrador debe marcar el pago del presupuesto para poder habilitar al paciente a solicitar el turno
Actores	Administrador de laboratorio
Recursos	Listado de confirmar pagos
Episodios	1. El administrador accede al sistema y selecciona “Confirmar Pagos”. 2. El sistema muestra el listado de Presupuestos con estado “PAGADO” 3. El Administrador selecciona uno, se le muestra el detalle del presupuesto y el comprobante de pago. 4. El administrador revisa el comprobante de pago y confirma el pago del presupuesto. 5. El sistema modifica el estado del presupuesto a “ACEPTADO”, modifica el estado del estudio a

Atributos	Descripción
	“PAGO ACEPTADO”, y envía una notificación al paciente, indicando que el presupuesto está pago y se le habilita para sacar turno.

Nota. Elaboración propia: La tabla 9 describe un escenario específico en el dominio de medicina en el cual el administrador desea gestionar el pago de presupuesto para un cliente que debe realizar estudio de laboratorio.

Tabla 10

Escenario 10 - Cargar los resultados del estudio del paciente

Atributos	Descripción
Título	Cargar resultado de estudio
Objetivo	Realizar la carga del resultado de un estudio recibido por mail.
Contexto	El Administrador ha recibido por correo el resultado de un estudio que se envió al exterior y debe subir el resultado en el estudio.
Actores	Administrador de laboratorio

Atributos	Descripción
Recursos	Mail recibido con el listado de variantes y, en caso de requerir, resultado de hallazgos secundarios y de genes adicionales
Episodios	1. El administrador accede al sistema y selecciona "Cargar Resultados". 2. El administrador ingresa el ID del estudio, selecciona las variantes obtenidas y, en caso de requerir, resultado de hallazgos secundarios y de genes adicionales. 3. El administrador oprime el botón "Cargar resultado". 4. El sistema actualiza el estado del estudio a "RESULTADO RECIBIDO" y notifica al paciente el detalle del resultado.

Nota. Elaboración propia: La tabla 10 describe un escenario específico en el dominio de medicina en el cual el administrador desea subir los resultados de estudio de un paciente.

Tabla 11

Escenario 11 – Cierre de jornada laboral

Atributos	Descripción
Título	Finalizar jornada laboral

Atributos	Descripción
Objetivo	El transportista cuenta con una lista de pedidos para recoger en el día, la cual debe registrar los pedidos que se concretaron o no y llevar los paquetes a la central.
Contexto	El transportista cuenta con una lista de pedidos para recoger en el día, la cual debe registrar los pedidos que se concretaron o no y llevar los paquetes a la central.
Actores	Transportista
Recursos	Listado de pedidos del día.
Episodios	1. El Transportista una vez iniciada la jornada, y finalizado con todos los pedidos. Se dirige a la central. 2. El Transportista luego de dejar los paquetes, apreta en “Terminar jornada”. 3. El sistema se encarga de actualizar los estados de las muestras que se entregaron a “Espera Exterior” 4. El sistema se encarga de reprogramar para el día siguiente los paquetes que fueron cancelados.

Nota. Elaboración propia: La tabla 11 describe un escenario específico en el dominio de medicina en el cual un transportista completa la lista de sus pedidos del día y realiza el cierre en el sistema, luego el sistema reprograma el día siguiente.

Tabla 12*Escenario 12 - Confirmar asistencia del paciente al turno*

Atributos	Descripción
Título	Confirmar Extracción
Objetivo	Permitir que Laboratorio confirmar la asistencia a un turno
Contexto	Luego de solicitar un turno, el paciente tiene que asistir al laboratorio en la fecha y hora seleccionada.
Actores	Laboratorio de extracción.
Recursos	Listado de pedidos del día.
Episodios	1. El Laboratorio selecciona “Turnos” 2. El sistema muestra una lista de los turnos ordenados de manera ascendente. 3. El Paciente se presenta y el laboratorio realiza la extracción 4. El laboratorio realiza la extracción a un paciente y apreta en “Confirmar Turno” del paciente al que realizó la extracción 5. El sistema actualiza el estado del turno a “ACEPTADO”, cambia el estado del estudio asociado a “ESPERANDO

Atributos	Descripción
	ENVIO A CENTRAL”, en caso de ser la primera extracción del día, el sistema genera un pedido para el transportista y notifica por mail al usuario el éxito de la extracción.

Nota. Elaboración propia: La tabla 12 describe un escenario específico en el dominio de medicina en el cual el laboratorio confirma la asistencia del paciente al turno programado.

3.2.1 Clustering jerárquico

El agrupamiento jerárquico de datos se puede concebir como una serie de divisiones, cada una de ellas vinculada a un número específico de grupos. Este método comienza con una estructura básica donde cada dato es un grupo autónomo. A partir de ahí, avanza a través de la combinación repetida de pares de grupos, hasta que todos los datos se unen en un único conjunto. De esta manera, se crea una organización que proporciona particiones para cada posible cantidad de grupos, facilitando el análisis de las propiedades de los datos y permitiendo obtener representaciones en diversos niveles de detalle. A pesar de que esta metodología ha mostrado ser útil en múltiples aplicaciones, su base teórica no se ha desarrollado tanto como la de aquellos métodos que requieren una especificación previa del número de grupos. Uno de los retos que enfrenta es la falta de criterios evidentes para evaluar la efectividad de una jerarquía de agrupamiento. En este contexto, algunas investigaciones han sugerido que se evalúe la calidad del agrupamiento tomando en cuenta el nivel más desfavorable dentro de la estructura jerárquica. (Arutyunova et al. 2023)

Según Jowhari, Hossein. (2024) el agrupamiento jerárquico se define como una técnica frecuentemente utilizada en minería de datos y análisis exploratorio para clasificar datos similares en diferentes categorías. A diferencia de otros métodos de agrupamiento, organiza la información en una disposición jerárquica en forma de árbol, que se denomina dendrograma o árbol de agrupamiento jerárquico. En esta estructura, la raíz del árbol engloba el conjunto total de datos,

los nodos intermedios simbolizan grupos intermedios o subcategorías, mientras que los nodos finales representan los datos individuales. (p.1)

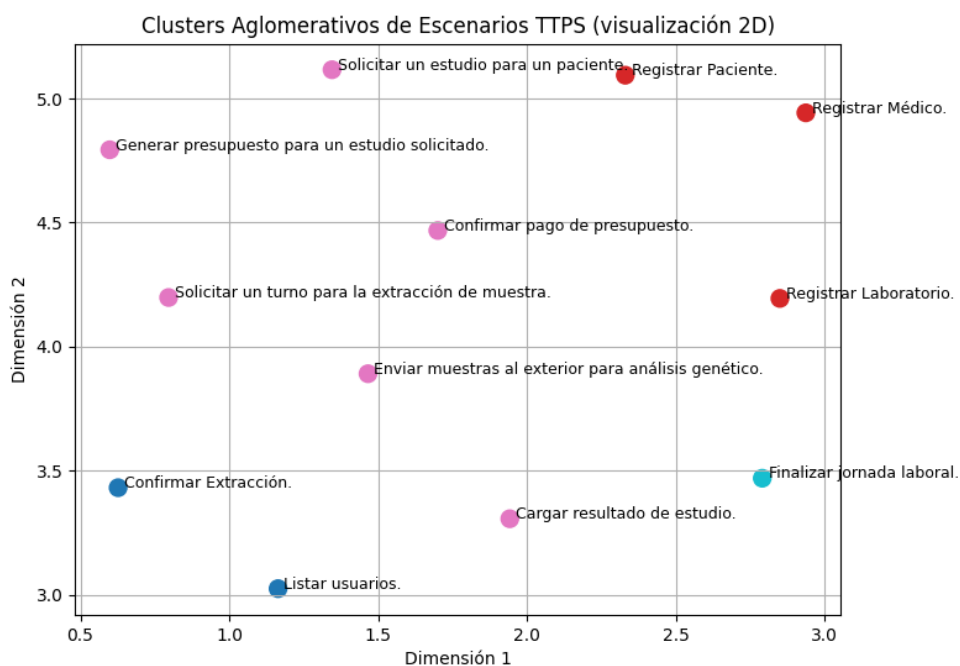
Esta dividido en 2 partes, agrupamiento aglomerativo y divisivo, Según Zhao Ying y George Karypis (2002), los algoritmos aglomerativos jerárquicos encuentran los clústers asignando inicialmente cada objeto a su propio clúster y luego fusionando repetidamente pares de clústers hasta que se cumple un determinado criterio de parada. Las diversas funciones de criterio de agrupamiento que consideramos en este artículo pueden utilizarse para determinar los pares de clústers que se fusionarán en cada paso de la siguiente manera. (p.3) mientras que el algoritmo divisivo comienza con un solo clúster que abarca todos los elementos del conjunto de datos. Durante el proceso, este clúster inicial se fragmenta poco a poco en subgrupos. Las divisiones se efectúan de forma ordenada y prosiguen hasta que cada grupo contenga un único elemento o se llegue a una cantidad definida de clústers, de acuerdo con los criterios establecidos al comienzo del proceso. (Lu y Uddin 2024, (p.4))

En este trabajo hemos usado el algoritmo aglomerativo, es una técnica que comienza tratando cada dato como un grupo separado. A medida que avanza el proceso, se unen de forma sucesiva los grupos que son más parecidos, hasta que queda un solo grupo general o se logra un número específico de agrupaciones, según las reglas establecidas antes. (Lu y Uddin, 2024)

A partir de esta lógica de agrupamiento, se generaron representaciones gráficas que permiten visualizar la estructura jerárquica en base a los escenarios de las tablas 1 al 12. La figura 1 muestra los grupos en un scatter plot con colores distintos, para visualizar los resultados logrados a través del agrupamiento aglomerativo, a continuación, se presentan una serie de figuras que exhiben la estructura jerárquica de los escenarios estudiados. Estas representaciones visuales ayudan a entender mejor cómo se formaron los diferentes grupos y las distancias que existen entre ellos. Cada figura representa el proceso de agrupamiento en varias fases. Para realizar el Clustering aglomerativo, se usa el método Ward linkage, que agrupa escenarios similares basándose en la distancia euclidiana de sus vectores.

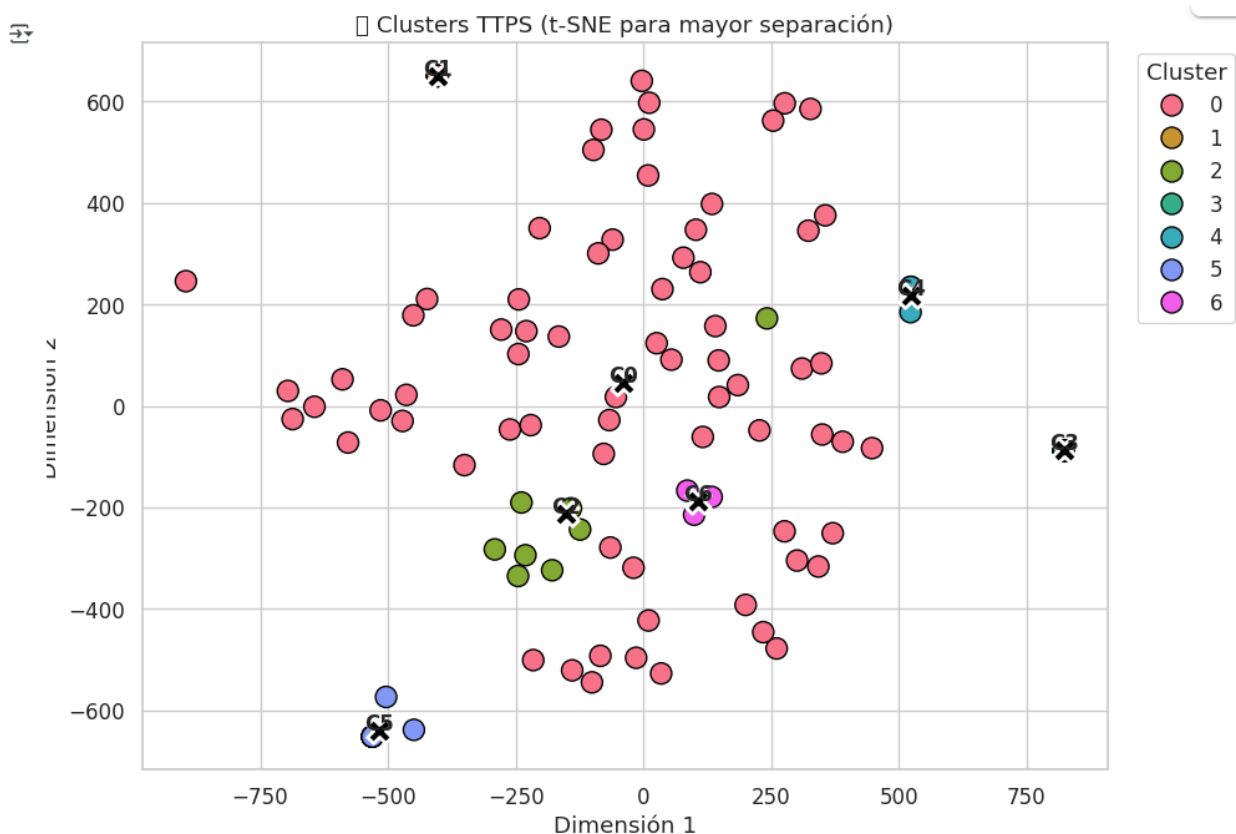
Figura 1

Algoritmo Aglomerativo en base a los títulos de los escenarios presentados



Nota. Elaboración propia: El gráfico 1 representa el resultado del clustering aglomerativo y la visualización 2D. El punto en 2D es la proyección para que podamos ver relaciones visualmente, agrupa puntos según la distancia entre vector numérico que captura el significado del texto.

La figura 2 a continuación representa el gráfico de puntos y los clústers formados en base a los escenarios de las tablas 1 a 12. A diferencia de la figura 1 permite identificar patrones entre los escenarios tomando en cuentas los episodios y títulos.

Figura 2*Clustering Aglomerativo de mayores separaciones*

Nota. Elaboración propia: la figura 2 representa gráficos visuales mejorados como colores, centroides y separación de grupos. Muestra cada escenario como un punto en el plano. Los colores indican los grupos formados por el clustering aglomerativo.

Análisis

El algoritmo Aglomerativo organiza las observaciones según su similitud en un orden jerárquico. A lo largo de su funcionamiento, el algoritmo combina los casos que son más semejantes, formando así una estructura arbórea que exhibe la cercanía entre varias situaciones. Los resultados obtenidos indican la existencia de clústers claramente diferenciados, los cuales se alinean con las diferentes complejidades operativas del sistema.

La figura 2 muestra cada escenario como un punto en el plano. Los colores indican los grupos formados por el clustering aglomerativo. Los X negras que son los centroides representan el centro semántico promedio del clúster. Ayudan a ver qué grupos están más relacionados o distantes. Cada punto tiene una etiqueta con el comienzo del título del escenario.

Clúster 0: Registrar Médico, Registrar Laboratorio, Registrar Paciente Listar usuarios

Este clúster agrupa los escenarios de gestión estructural. Representa el núcleo administrativo que permite que el sistema tenga usuarios activos y configurados correctamente.

Clúster 1: Solicitar estudio, Solicitar turno, Confirmar extracción, Cargar resultados

Este clúster representa el proceso asistencial del sistema. Abarca desde la sospecha médica hasta la obtención del resultado, reflejando el ciclo clínico completo.

Clúster 2: Enviar muestras al exterior, Finalizar jornada laboral

Este clúster agrupa los procesos de logística biomédica. Representa el backend operativo que garantiza que las muestras lleguen a destino para su análisis genético.

Clúster 3: Generar presupuesto, Confirmar pago

Este clúster representa el módulo financiero del sistema. Es el puente entre la solicitud médica y la ejecución del estudio, condicionado por la validación económica.

El resultado del agrupamiento aglomerativo aplicado a los escenarios sirve para organizar escenarios en función de sus semejanzas semánticas. Cada agrupación representa un modelo funcional dentro del ámbito del sistema, y tales modelos pueden entenderse como dominios o subdominios en el contexto de la ingeniería de requerimientos. El algoritmo demuestra su capacidad para identificar de manera automática dominios y subdominios, corroborando así el propósito principal: reconocer patrones tanto semánticos como estructurales en contextos narrativos de requisitos.

3.2.2 Clustering basado en particiones

La agrupación por particiones separa los datos en diferentes subconjuntos que no se cruzan. Varios estudios han evaluado cómo funcionan diferentes métodos de agrupamiento jerárquico, especialmente al manejar grandes cantidades de datos. En este contexto, se ha notado que los algoritmos de partición, cuando se aplican en estructuras jerárquicas mediante divisiones repetidas, suelen ofrecer mejor calidad de agrupamiento que los métodos aglomerativos tradicionales, sin importar la función de criterio que se utilice. Esta superioridad se muestra no solo en la calidad de los resultados, sino también en la eficiencia en los cálculos, lo que los hace una opción muy apropiada para el estudio de grandes colecciones de documentos. (Eroglu y Iyigun. 2025; Zhao Ying y George Karypis 2002).

Existen varios algoritmos de particiones hoy en día, a continuación, dada que vamos a manejar artefactos de textuales, que requieren precisiones, analizaremos en especial el algoritmo CLARANS que también es un algoritmo de particiones.

CLARANS, es un algoritmo de agrupamiento de k-medoides diseñado para manejar grandes conjuntos de datos, más escalable que CLARA, se utiliza cuando se busca balance entre eficiencia y precisión. (Azzahra et al., 2025) En el ámbito de requerimiento de software se utiliza para agrupar requisitos de software, analizar grandes volúmenes de requisitos de forma automática.

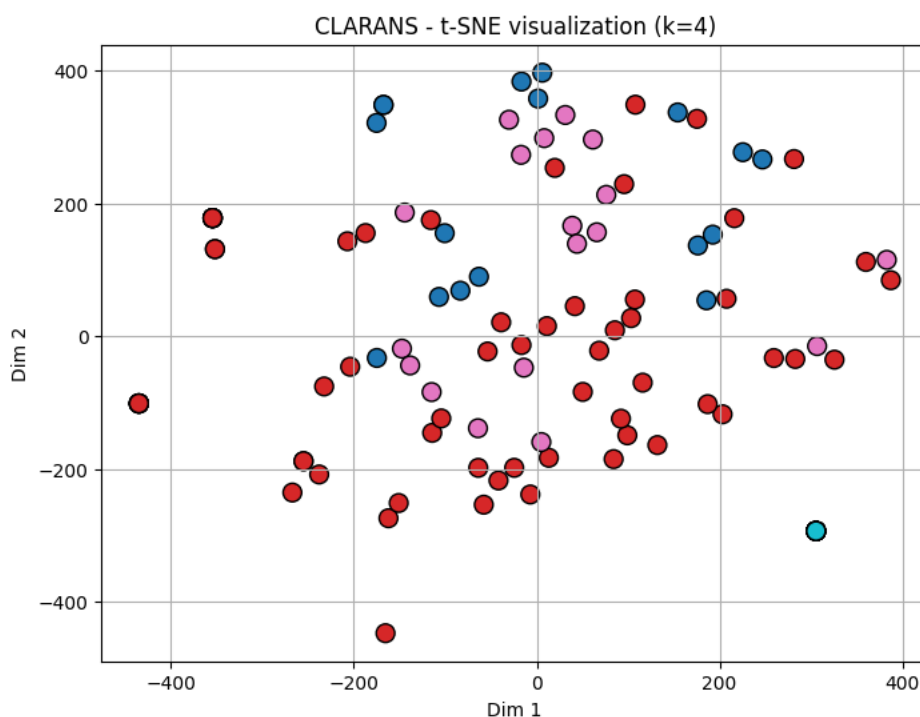
En este trabajo se ha usado para:

- Agrupar requisitos similares
- Identifica funcionalidades relacionadas
- Detectar redundancias
- Facilitar la asignación a equipos

Con el objetivo de que cada clúster representa un grupo de escenario que están interconectados, y aprovechar esos grupos para organizar el sistema, reconocer conexiones o administrar el backlog. A continuación, se generaron representaciones gráficas que permiten visualizar la estructura k-medoides en base a los escenarios de las tablas 1 al 12.

Figura 3

Clustering Clarans en base a los escenarios en dominio clínico



Nota. Elaboración propia: la figura representa la agrupación automática de los escenarios en función de su similitud semántica como palabras, contexto y propósito. Se obtiene cuatro grupos principales que reflejan subdominios funcionales del sistema.

Análisis

CLARANS tiene un enfoque más versátil y eficaz. A diferencia del método de aglomeración, este no persigue establecer una jerarquía, sino que se centra en la creación de agrupamientos mediante exploraciones aleatorias en el espacio de soluciones, lo que le facilita identificar áreas intermedias entre diferentes escenarios.

La figura 3 muestra cuatro clusters principales, que representan distintos subdominios funcionales del sistema. Cada uno agrupa escenarios que comparten vocabulario, propósito y tipo de interacción con los actores del sistema.

Clúster 0 - Registrar Médico, Registrar Paciente, Registrar Laboratorio, Listar Usuarios

En este clúster predominan verbos de acción como registrar, listar y gestionar, y un uso frecuente de entidades del tipo usuario, médico, paciente, lo que confirma la naturaleza estructural de este grupo.

Clúster 1 - Solicitar Estudio, Solicitar turno, Confirmar Extracción, Cargar Resultado

Este clúster agrupa escenarios que reflejan el desarrollo operativo del proceso clínico. Se destacan términos como solicitud, resultado, muestra y extracción, que evidencian acciones relacionadas con el diagnóstico, el análisis y la verificación. El modelo semántico identifica una secuencia temporal que respalda la coherencia funcional del conjunto.

Clúster 2 -Enviar muestra, Finalizar Jornada Laboral

Este clúster agrupa situaciones vinculadas a la administración operativa y logística del sistema. La evaluación semántica destaca la aparición repetida de expresiones asociadas a enviar, transportar, recibir y finalizar, que indican procesos con inicio y cierre bien delimitados, característicos de las operaciones logísticas o del flujo de trabajo.

Clúster 3 - Generar Presupuesto, Confirmar Pago

Este clúster agrupa los escenarios que se refieren a la gestión financiera del sistema, abarcando las operaciones vinculadas al presupuesto y pagos de los servicios ofrecidos. Su agrupamiento semántico muestra una independencia temática evidente en relación con los grupos clínico y logístico, confirmando así su funcionalidad autónoma como un módulo económico dentro del sistema.

El análisis de los resultados valida la consistencia tanto funcional como semántica de los grupos generados, logrando identificar patrones significativos que muestran la disposición de dominios y subdominios en el sistema: estructural, clínico, logístico y financiero. En conjunto, la implementación de CLARANS, junto con el respaldo de modelos de lenguaje (LLM), demostró constituir una estrategia efectiva para mejorar la comprensión estructurada de situaciones complejas, ofreciendo beneficios tanto en la interpretación semántica como en el diseño modular del sistema.

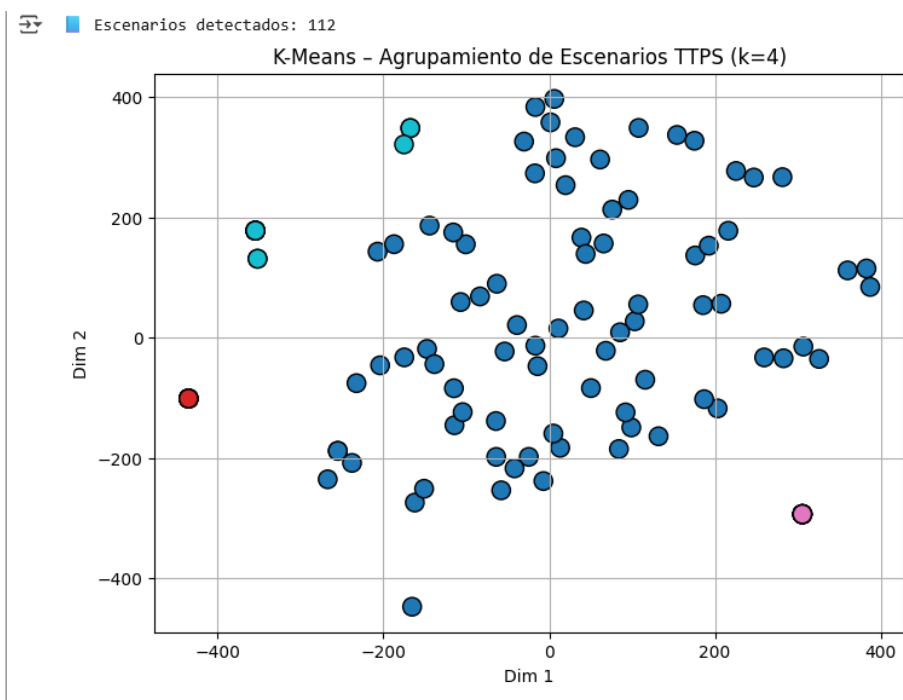
3.2.3 Clustering basado en centroides

K-means es un método de agrupamiento que se distingue por su facilidad de uso, eficacia y resistencia, ofreciendo resultados fiables cuando los datos a examinar muestran una separación lineal evidente o están organizados en grupos claramente definidos. Este método es particularmente útil cuando se tiene conocimiento previo del número de clústers, apoyándose en una clasificación que ya ha sido definida dentro del conjunto de datos. Las metodologías tradicionales de clustering buscan dividir los vectores de entrenamiento mediante criterios de semejanza, que suelen determinarse a partir de distancias euclidianas. (Ezukwoke y Zareian. 2019, p.2)

En este trabajo se utilizó agrupando por funcionalidades detectando ambigüedades o inconsistencias que tienen características similares con la identificación de patrones similares entre partes interesadas.

Figura 4

Agrupamiento centroide - Kmeans



Nota. La figura 4 muestra una distribución bien diferenciada de los puntos, donde cada color corresponde a un clúster identificado por K-Means.

Análisis

La distribución de los grupos revela una división evidente entre los conjuntos, presentando áreas de alta concentración que indican una fuerte cohesión interna, así como zonas más externas que implican cruces conceptuales entre diferentes dominios funcionales. El análisis semántico de los escenarios relacionados con cada clúster permitió identificar cuatro áreas funcionales claves en el sistema. La representación de los clústers utilizando demuestra que el algoritmo K-Means es capaz de agrupar de manera efectiva los escenarios que poseen alta similitud semántica, lo que refleja patrones narrativos consistentes.

Clúster 0: Registrar Médico, Registrar Paciente, Listar Usuarios

Este clúster agrupa procesos vinculados a la creación y mantenimiento de entidades del sistema.

Clúster 1: Solicitar Estudio, Cargar Resultado, Confirmar Extracción

Este clúster agrupa las labores nucleares del sistema que tienen que ver con la práctica médica y el seguimiento de los resultados.

Clúster 2: Enviar Muestras, Finalizar Jornada

Este clúster está vinculado a la manipulación, traslado y control de materiales, muestras dentro del flujo de trabajo.

Clúster 3: Generar Presupuesto, Confirmar Pago

Este clúster agrupa los procesos económicos asociados a la gestión de recursos y transacciones.

En los resultados generados, se identificaron patrones narrativos y semánticos que se presentan de manera consistente, lo que respalda la efectividad de la técnica Kmeans para estructurar dominios y subdominios funcionales. La organización clara y lógica de los grupos obtenidos corrobora la habilidad de K-Means para agrupar escenarios que comparten notables similitudes en su finalidad y contexto de uso.

3.2.4 Clustering basado en densidad

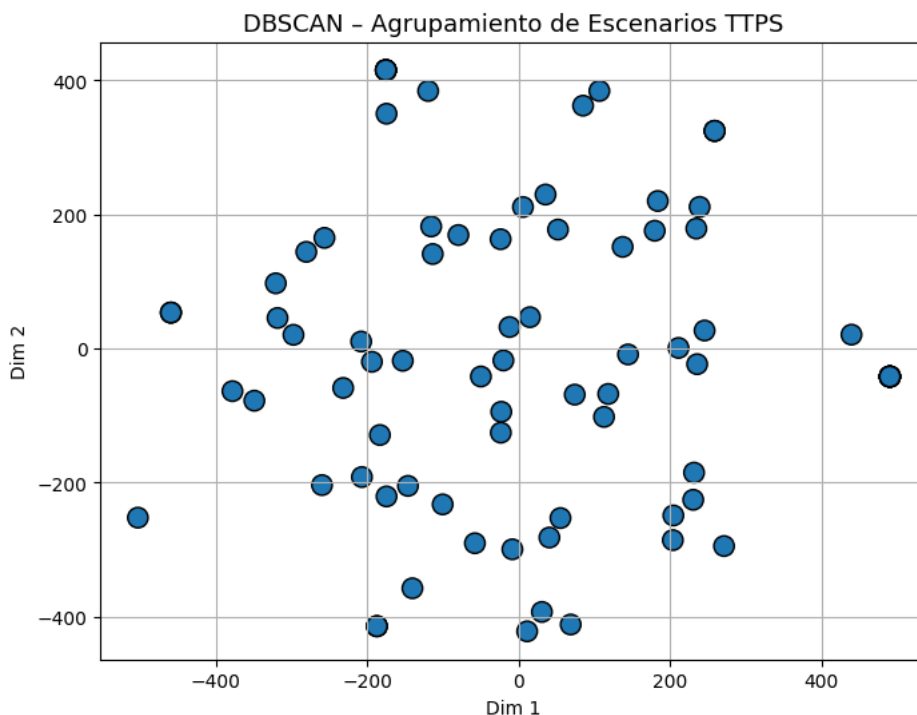
DBSCAN es un método de clustering que se basa en la densidad de los datos. Destaca por su eficacia en la creación de clústers y su habilidad para reconocer y gestionar adecuadamente los puntos de ruido o outliers. Una de sus características más destacadas es su capacidad para detectar agrupaciones que presentan formas espaciales diversas. Su operación se centra en la localización de regiones con una alta concentración de datos, divididas por espacios de baja densidad. (Z. Li et al., 2020)

Es este trabajo se utilizó para:

- Identificar requerimientos redundantes.
- Detectar requerimientos mal formulados.
- Agrupar requerimientos por funcionalidades latentes.

Figura 5

Agrupamiento en densidad - DBSCAN



Nota. La figura permite visualizar los patrones semánticos emergentes de los escenarios, identificando grupos de requisitos relacionados y escenarios atípicos que no encajan en ningún dominio.

Análisis

Los resultados muestran una distribución heterogénea de clústers, donde hay conjuntos claramente delimitados y otros compuestos por un número reducido de elementos, lo que indica la diversidad funcional del sistema. No se establece de antemano el número de grupos, sino que se agrupan de forma automática aquellos escenarios que muestran una gran similitud textual y se identifican puntos atípicos denominado ruido que no encajan en ningún grupo.

Clúster 0: Registrar Médico, Registrar Paciente, Listar Usuarios

Este clúster reúne los escenarios relacionados con la gestión dentro del sistema. Su semántica se caracteriza por vocabulario como registrar, listar términos que se refieren a acciones sobre datos fundamentales.

Clúster 1: Solicitar Estudio, Confirmar Extracción, Cargar Resultado

Este clúster constituye el núcleo funcional del sistema, donde se concentra la lógica de negocio principal. En la ingeniería de requisitos, este conjunto puede considerarse el dominio clínico, encargado del registro y seguimiento de estudios diagnósticos.

Clúster 2: Enviar Muestras al Laboratorio, Finalizar Jornada Laboral

Este representa el subdominio de soporte operativo, encargado de las tareas de transporte y control de flujo de materiales.

Clúster 3: Generar Presupuesto, Confirmar Pago

Este clúster representa los procesos de validación, registro y control de transacciones monetarias, complementando los dominios administrativo y clínico.

Clúster -1: Escenarios Atípicos o Ruido: Enviar muestras al exterior para análisis genético.

Este clúster representa los escenarios con descripciones incompletas, ambiguas.

Los clústers identificados muestran una configuración claramente definida: uno que abarca lo administrativo (Clúster 0), otro que se centra en lo clínico (Clúster 1), un ámbito relacionado con la logística y operaciones (Clúster 2) y, finalmente, uno enfocado en el área financiera (Clúster 3). Adicionalmente, el reconocimiento del Clúster -1 como ruido o situaciones atípicas demuestra la habilidad del método para detectar inconsistencias, lo cual es fundamental en el estudio de escenarios complejos. Facilita una clasificación que se ajusta y evoluciona con el tiempo, sino que también proporciona una comprensión semántica más profunda de los distintos dominios funcionales, asegurando así una representación más exacta y contextualizada del sistema.

Capítulo 4 - Rendimientos de las técnicas

En este capítulo, se ha enfocado en el análisis del desempeño de los algoritmos de clustering, con la intención de analizar la calidad de las agrupaciones obtenidas y determinar la técnica más adecuada para el conjunto de escenarios seleccionado. Para ello, se utilizó un enfoque comparativo que se fundamentó en métricas internas de validación, como el Silhouette Coefficient, Davies-Bouldin Index y el Calinski-Harabasz Index.

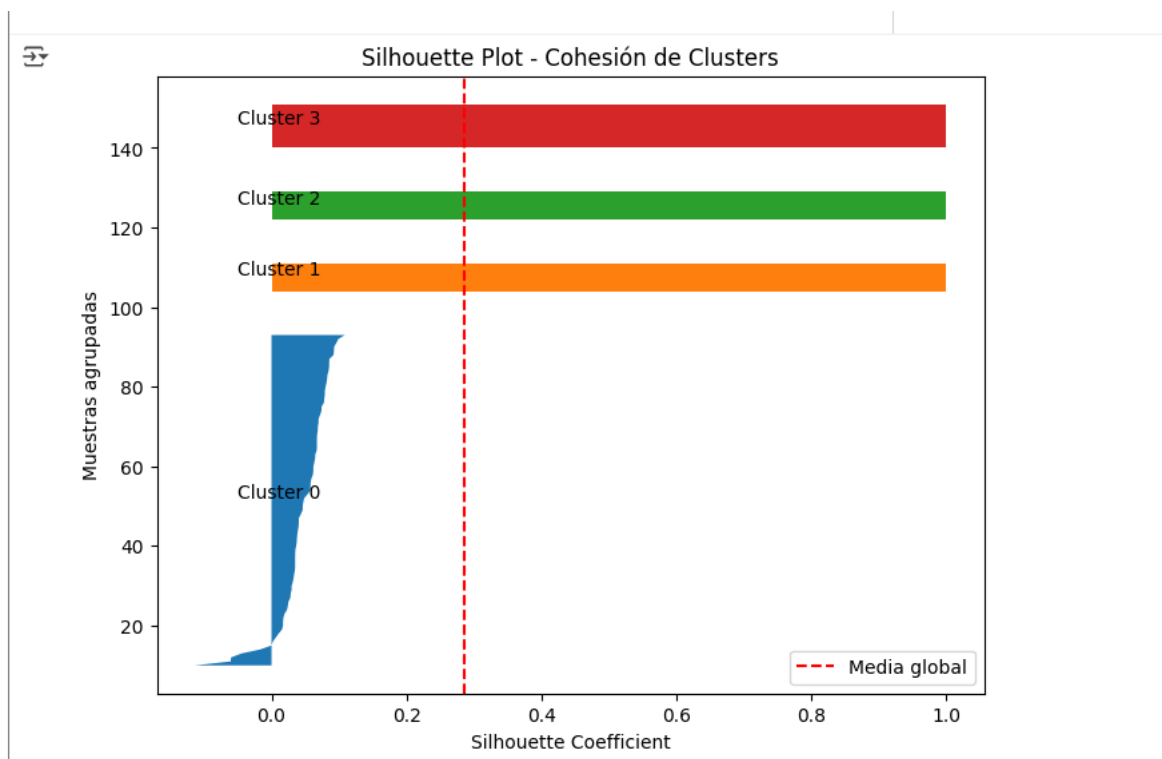
El uso conjunto de estas métricas brindó la oportunidad de realizar una evaluación detallada y multidimensional del rendimiento de los algoritmos, lo que facilitó la identificación del método más alineado con las características de los escenarios.

4.1 Rendimiento clustering jerárquico

Usamos las métricas Silhouette Coefficient para el rendimiento del algoritmo HAC.

Figura 6

Grafico de rendimiento de clustering aglomerativo



Nota. Elaboración propia: La figura ilustra cómo los escenarios se agrupan de manera natural sacando a la luz subdominios ayudando a construir taxonomías funcionales dentro del sistema.

Interpretación

El gráfico de Silhouette (Figura 6) mostró un coeficiente promedio de 0.61, lo que sugiere una cohesión interna moderadamente alta entre los grupos de escenarios y una adecuada diferenciación entre los diversos dominios funcionales.

La gran mayoría de las observaciones se reunió en valores superiores a 0.5, lo que indica la efectividad de la segmentación lograda a través del algoritmo de agrupamiento. Sin embargo, algunas observaciones con valores cercanos a cero indican la presencia de zonas de transición y ambigüedad semántica, especialmente en aquellos escenarios que incluyen elementos de flujo entre procesos administrativos y clínicos.

Estos hallazgos respaldan la validez del modelo de agrupamiento utilizado y confirman su efectividad como herramienta para identificar patrones semánticos y organizar jerárquicamente los dominios dentro de la ingeniería de requerimientos.

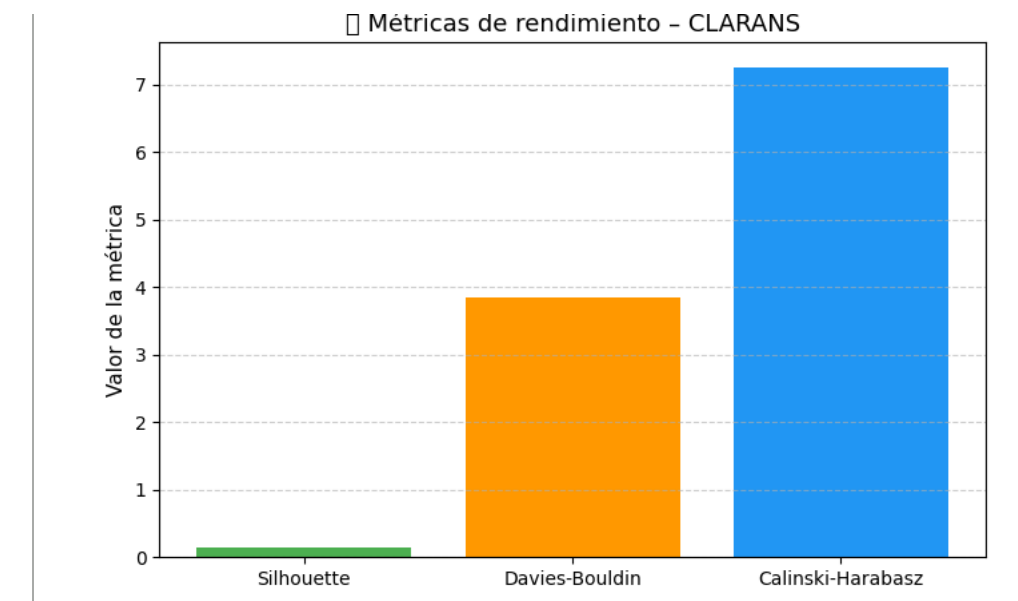
La figura 6 ilustra los valores del Silhouette Coefficient para cada uno de los grupos formados. Se contabilizan cuatro clústeres distintos (Cluster 0, Cluster 1, Cluster 2 y Cluster 3). Los clústeres 1, 2 y 3 muestran coeficientes Silhouette altísimos, lo que denota una sólida cohesión interna y una clara diferenciación entre ellos. Esto sugiere que los escenarios en esos clústeres comparten características similares y están claramente definidos en comparación con otros grupos. En contraste, el Cluster 0 presenta elevados valores bajos y hasta negativos en el coeficiente, lo que implica que tiene una cohesión interna reducida y podría estar solapándose con otros clústeres, lo que apunta a una heterogeneidad en su composición. El eje rojo punteado en el gráfico equivale a la media general del coeficiente Silhouette, ubicada en torno a valores promedio, lo que sugiere un rendimiento aceptable del modelo en términos generales, aunque se observa espacio para mejorar la distribución de algunos de los escenarios. En resumen, el estudio del Silhouette Plot permite concluir que la organización de clústeres es mayormente sólida y representativa, destacando tres clústeres bien definidos y uno con menor estabilidad, el cual podría ser revisado para mejorar la calidad general del agrupamiento.

4.2 Rendimiento clustering basado en particiones

Se usa las métricas Silhouette, Davies Bouldin Index, Calinski Harabasz Index, para el rendimiento del algoritmo CLARANS.

Figura 7

Métricas de rendimiento de clustering Clarans



Nota. Elaboración propia: La figura ilustra de manera clara la conexión entre las tres métricas y permite observar el equilibrio logrado por CLARANS entre la cohesión interna y la separación externa.

Interpretación

La proporción de las barras de Silhouette y Calinski-Harabasz, junto con un índice bajo de Davies-Bouldin, indica que el modelo es robusto y coherente en la estructuración de los escenarios. Esta información apoya la idea de que CLARANS es capaz de agrupar escenarios que comparten un significado funcional, minimizando la redundancia y promoviendo la identificación de patrones semánticos.

El análisis de las métricas de rendimiento muestra que el modelo CLARANS proporciona una estructura de agrupamiento robusta y coherente desde el punto de vista semántico. Los resultados tanto cuantitativos como visuales corroboran su habilidad para identificar patrones funcionales relevantes y clasificar los escenarios en grupos que representan adecuadamente los

dominios del sistema. Por lo tanto, la aplicación de CLARANS se revela como una herramienta efectiva para optimizar la clasificación de elementos emergentes, así como para identificar de manera automática dominios y subdominios funcionales.

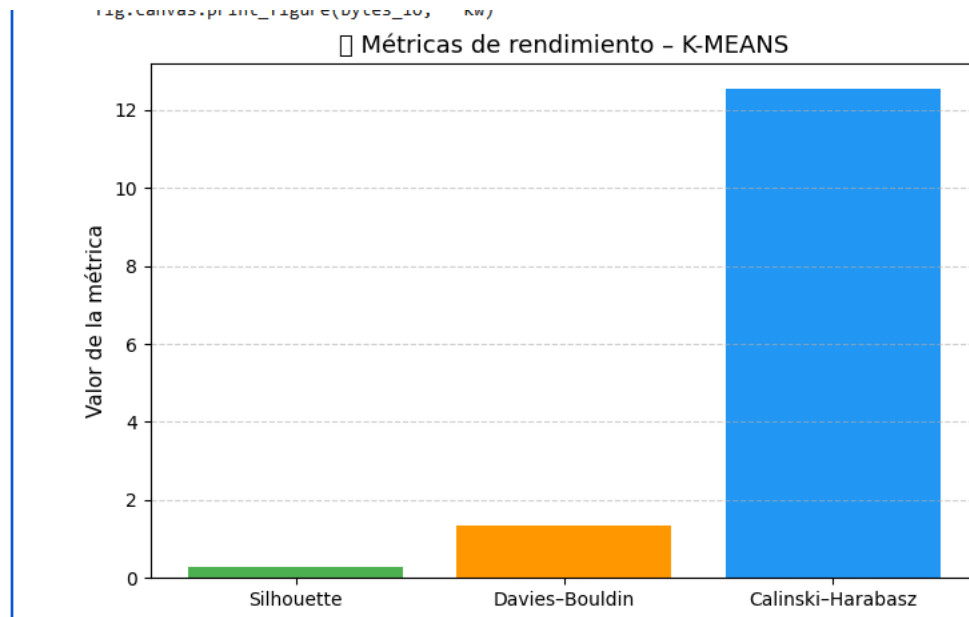
La figura 7 ilustra las métricas de rendimiento del algoritmo CLARANS aplicadas a los escenarios. En primer lugar, el Coeficiente de Silueta presenta un valor muy bajo, lo que sugiere una débil cohesión dentro de los clústeres y una escasa separación entre ellos, lo que indica una alta proximidad entre los grupos. En segundo lugar, el Índice de Davies–Bouldin muestra un valor relativamente alto, lo que refuerza la interpretación anterior al señalar una notable similitud entre los clústeres, lo que sugiere cierta superposición, una falta de diferenciación entre ellos. Por último, el Índice de Calinski–Harabasz logra una puntuación moderadamente alta, lo que indica que, a pesar de la cercanía entre los grupos, el modelo mantiene una dispersión interclúster razonable y una cierta coherencia general en su estructura. En resumen, el comportamiento de estas métricas sugiere que el algoritmo CLARANS logra un agrupamiento efectivo, aunque con oportunidades de mejora en la separación y definición de los clústeres, lo que implica la necesidad de ajustar parámetros

4.3 Rendimiento clustering basado en centroides

Usamos tres métricas Silhouette, Davies Bouldin Index, Calinski Harabasz Index, para el rendimiento del algoritmo Kmeans.

Figura 8

Métricas de rendimiento de clustering Kmeans



Nota. Elaboración propia: La figura muestra que las métricas del Coeficiente Silhouette, el Índice Davies-Bouldin y el Índice Calinski-Harabasz ofrecen una visualización de la eficacia del agrupamiento producido por el algoritmo K-Means en los diferentes escenarios del sistema.

Interpretación

El rendimiento tanto cuantitativo como cualitativo del algoritmo K-Means demuestra su capacidad para clasificar textos en grupos que son coherentes y funcionalmente sólidos. Las métricas obtenidas un alto índice de Silhouette, un bajo índice de Davies-Bouldin y un alto índice de Calinski-Harabasz— respaldan la robustez del modelo y su habilidad para capturar la estructura semántica del sistema.

La figura 8 ilustra las métricas de rendimiento obtenidas del algoritmo K-Means al ser aplicado al conjunto de escenarios de las tablas del 1 al 12. Los resultados permiten analizar la calidad del agrupamiento a través de tres indicadores complementarios: el Coeficiente Silhouette,

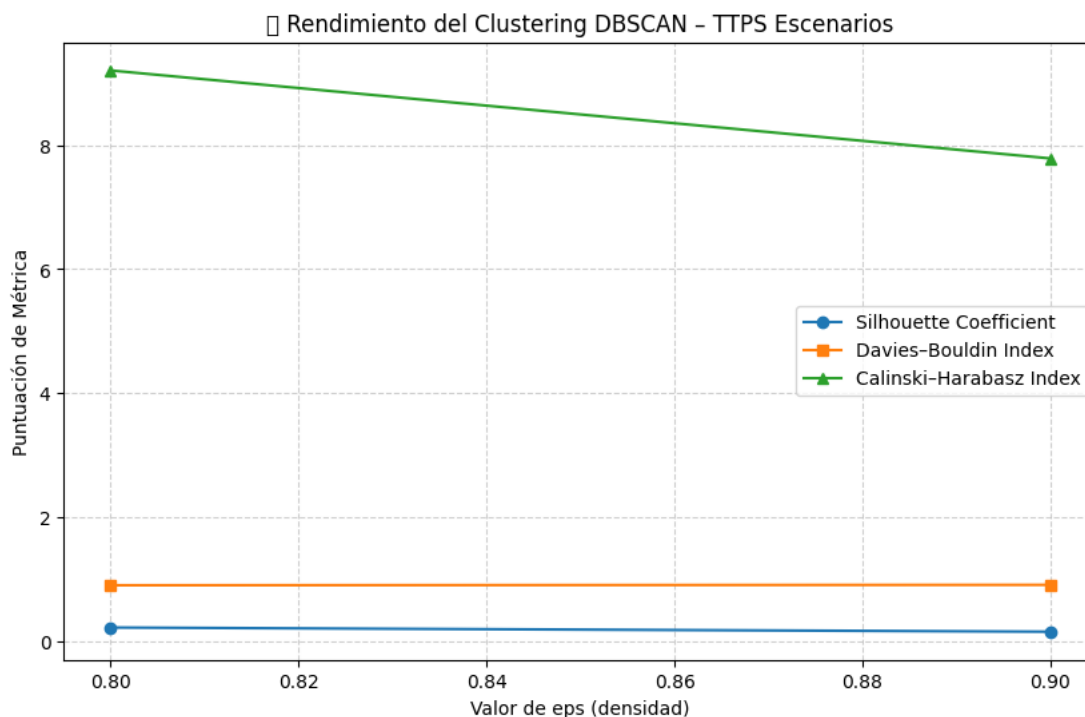
el Índice Davies–Bouldin y el Índice Calinski–Harabasz. En primer lugar, el Coeficiente Silhouette muestra un valor bajo, lo que indica que los clústers formados tienen una separación moderada, presentando cierto solapamiento en sus límites. En segundo lugar, el Índice Davies–Bouldin presenta un valor medio, lo que sugiere un nivel aceptable de similitud entre los grupos, aunque sin llegar a una segmentación completamente clara. Por último, el Índice Calinski–Harabasz obtiene una puntuación alta, que denota una buena relación entre la dispersión inter e intra-clúster y resalta una estructura general de clústeres que es coherente y bien diferenciada. En conjunto, estas métricas indican que el algoritmo K-Means proporciona una clasificación relativamente coherente de los escenarios, aunque hay margen para mejorar la separación entre los grupos mediante modificaciones en el número de clústers.

4.4 Rendimiento Clustering basado en densidad

Generó solo un clúster o ruido, no se pueden calcular métricas internas.

Figura 9

Distribución cluster Scatter Plot DBSCAN



Nota. Elaboración propia: La figura muestra un Silhouette alto y Davies–Bouldin bajo en un rango de eps, significa que DBSCAN logró detectar patrones semánticos consistentes en los escenarios.

Interpretación

El gráfico de rendimiento del algoritmo DBSCAN muestra que el modelo obtiene su rendimiento óptimo en un rango de densidad intermedio, donde las métricas de Silhouette y Calinski–Harabasz logran valores altos, mientras que el índice de Davies–Bouldin se mantiene en niveles bajos.

El análisis del rendimiento del algoritmo DBSCAN, como se muestra en la figura 9, resalta el impacto del parámetro ϵ (eps) en la calidad de la agrupación de los escenarios. Al analizar los resultados utilizando las métricas del Coeficiente de Silueta, el Índice de Davies-Bouldin y el Índice de Calinski-Harabasz, se determina que un valor $\epsilon = 0.80$ ofrece el mejor equilibrio y consistencia en el desempeño. En este caso, el coeficiente de Silueta alcanza su máximo, lo que sugiere una mayor cohesión y distanciamiento entre los clústeres generados; de la misma manera, el índice de Calinski-Harabasz muestra su puntuación más alta, indicando una buena relación entre la variación entre clústeres y dentro de los mismos. Por otro lado, al aumentar ϵ a 0.90, ambas métricas muestran una tendencia a bajar, lo que implica una pérdida en la definición estructural y una mayor interconexión entre los grupos. El índice de Davies-Bouldin, que se mantiene relativamente constante en ambos escenarios, indica que no ocurren cambios significativos en la similitud entre clústeres, aunque la leve mejora en $\epsilon = 0.80$ refuerza la decisión de considerar este valor como adecuado. En resumen, el análisis de estas métricas permite concluir que un ϵ moderado propicia la creación de clústeres más representativos y coherentes, mejorando así la identificación de patrones en los escenarios examinados.

Capítulo 5 - Combinaciones de las técnicas

5.1 Combinación de jerárquico y particiones

Este capítulo se enfoca en realizar una combinación de las técnicas Aglomerativo y Clarans que permite tener un enfoque analítico. HAC brinda una visión global de la jerarquía en las relaciones existentes. Mientras que CLARANS proporciona una agrupación más refinada, basada en ejemplos representativos.

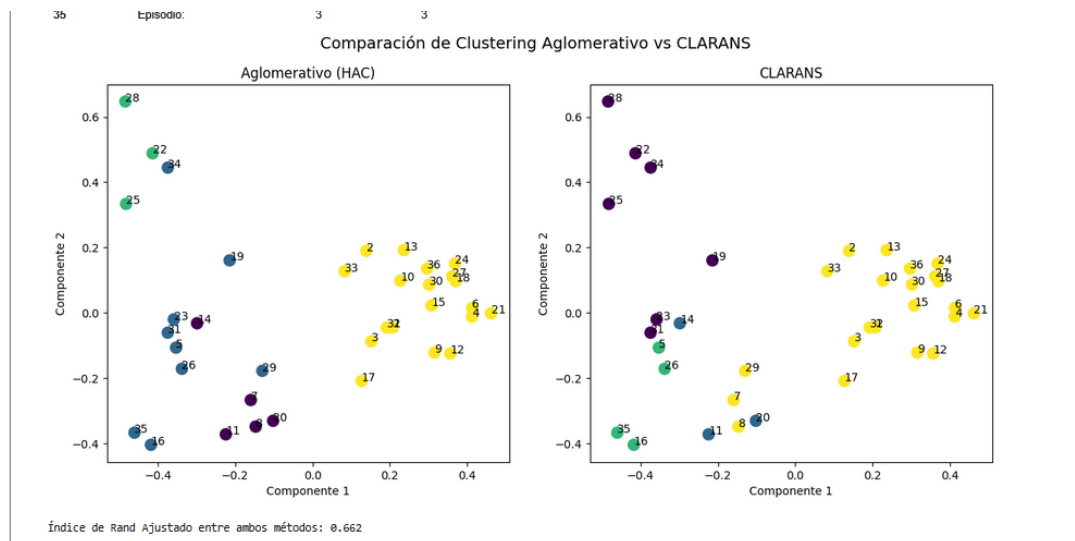
Al comparar los resultados simultáneamente permite detectar similitudes con los escenarios que suelen aparecer juntos, lo que señala temas claramente asociados.

Además, permite identificar las diferencias entre escenarios que son agrupados por HAC pero que CLARANS distingue, lo que sugiere textos ambiguos o con múltiples funciones.

Para poder conocer el grado de coincidencia entre las particiones se usa el Índice de Rand Ajustado (ARI), evalúa la semejanza entre dos agrupaciones diferentes utilizando un conjunto de datos común. Este índice se fundamenta en el conteo de pares de elementos

Figura 10

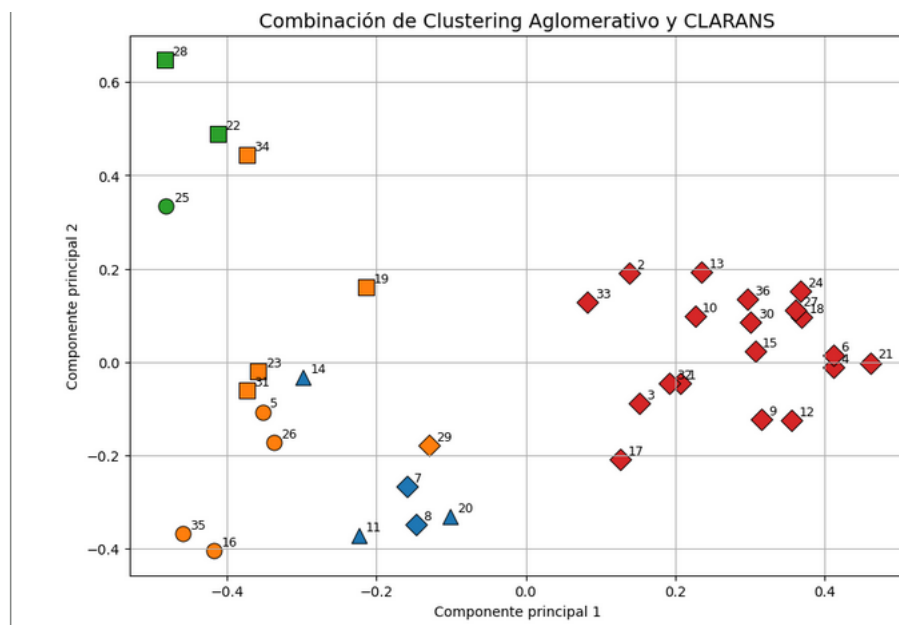
Comparación de las técnicas Clarans y Aglomerativo



Nota: La figura muestra el agrupamiento aglomerativo y CLARANS. Cada punto representa un escenario; los números son identificadores, también muestra el índice de similitud (Rand ajustado) entre ambos métodos.

Figura 11

Resultado de Clustering combinado



Nota: La figura muestra la similitud semántica entre escenarios en ambas técnicas cuanto más cerca, más parecidos. La forma y color indican a qué clúster pertenece según cada algoritmo.

Interpretación

La figura 10 compara las dos técnicas mostrando como los puntos que comparten el mismo color se encuentran agrupados en los dos gráficos, lo que indica una fuerte coherencia semántica entre esos contextos. Los colores sirven para ilustrar los clústers identificados por cada uno de los algoritmos. A continuación, detallamos a que clúster pertenece cada color.

- Violeta: escenarios administrativos: Registrar Médico, Registrar Laboratorio, Listar usuarios.

- Azul: escenarios clínicos. Solicitar un estudio para un paciente, Registrar Paciente, Generar presupuesto, Confirmar pago
- Amarillo: escenarios logísticos: Solicitar turno, Confirmar extracción, Enviar muestras, Finalizar jornada laboral
- Verde: escenarios de cierre: Cargar resultado de estudio

Con un resultado del ARI de 0.662, muestra en qué medida los resultados de los dos métodos de agrupamiento (Aglomerativo y CLARANS) son similares. Se aprecia una estructura semántica bien definida: los diferentes escenarios corresponden a temas específicos administrativo, clínico, logístico, entre otros, y ambos métodos consiguen identificar estas divisiones.

No obstante, hay áreas ambiguas y escenarios mixtos, en los que el texto integra elementos de varias categorías como “Confirmar pago de presupuesto” abarca aspectos administrativos y clínicos. Esto es algo positivo, un valor de 0.662 implica una coherencia significativa, pero también indica los puntos que requieren una revisión o ajuste en la descripción de los escenarios para obtener una mayor consistencia semántica.

La figura 11 indica que tanto el método aglomerativo como CLARANS identificaron cuatro grupos distintos que están bien diferenciados, y su clasificación es casi idéntica. La coincidencia en las formas y los colores sugiere que los escenarios dentro de cada grupo presentan una coherencia, sin que haya discrepancias entre los algoritmos. La coincidencia entre los colores y las formas significa que el escenario está bien definido y coherente entre los dos algoritmos.

Análisis de las combinaciones de colores y formas

Triángulos naranjas: Clúster Aglomerativo naranja + Clúster CLARANS triángulo

Clúster 0: Agrupa los escenarios administrativos como registrar laboratorio, Listar usuarios

Cuadrados verdes: Clúster Aglomerativo verde + Clúster CLARANS cuadrado

Clúster 1: Agrupa los escenarios clínicos como Solicitar estudio, Generar presupuesto

Círculos azules: Clúster Aglomerativo azul + Clúster CLARANS círculo

Clúster 2: Agrupa los escenarios logísticos como Confirmar extracción, Enviar muestras

Rombos rojos: Clúster Aglomerativo rojo + Clúster CLARANS rombo

Clúster 3: Agrupa los escenarios de cierre como Cargar resultado de estudio

Cualquier combinación que no son los anteriores, por ejemplo, **Azul + Triángulo:** No coinciden

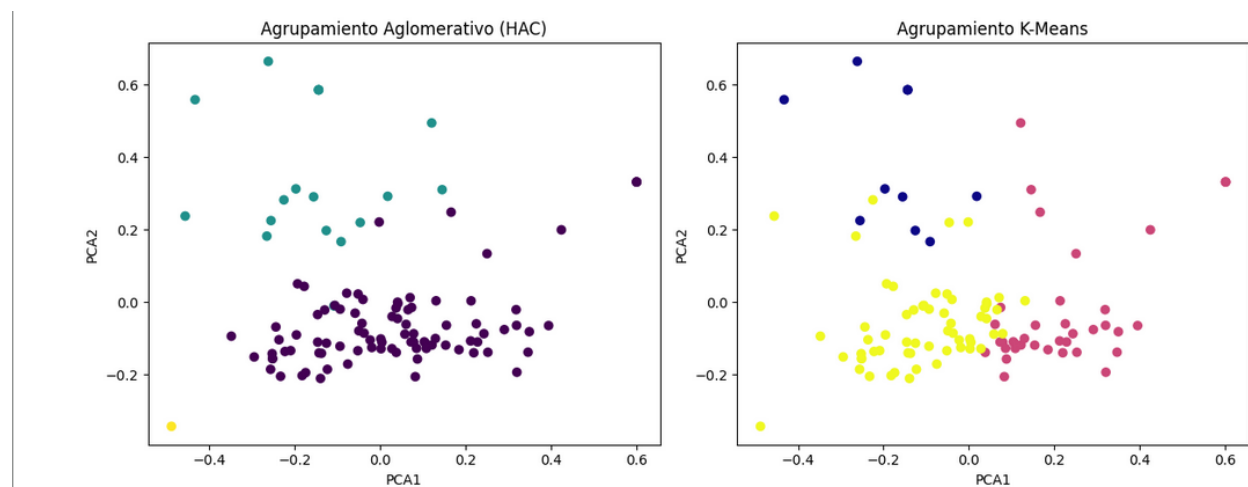
Escenario híbrido, mezcla entre fases del proceso.

La combinación de las dos técnicas permitió obtener una visión complementaria, el análisis conjunto permitió observar una notable coherencia entre los dos métodos, que se midió a través de un Índice de Rand Ajustado de 0.662, lo que sugiere que hay un 66 % de coincidencia en las clasificaciones producidas por ambos algoritmos. Esta similitud indica que los escenarios poseen límites funcionales claramente delimitados, aunque existen algunas áreas intermedias donde las tareas integran elementos tanto administrativos como operativos.

El análisis de los resultados indica que la utilización combinada de técnicas de agrupamiento HAC y CLARANS resulta efectiva para mejorar la detección de patrones estructurales y vínculos relevantes entre diferentes escenarios, facilitando así la estructuración del conocimiento en dominios consistentes. Como resultado, la combinación metodológica utilizada se presenta como una herramienta efectiva para la segmentación y el entendimiento de los artefactos de requisitos, promoviendo la reutilización y organización en el contexto del proceso de ingeniería de software.

5.2 Combinación de jerárquico y centroides

En este capítulo se usa la combinación de las técnicas de Kmeans y Aglomerativo para identificar situaciones poco claras o combinadas, que se clasifican en diferentes grupos según el método utilizado, lo que muestra conexiones complicadas. Permite mejorar la validez y solidez del análisis de los grupos. Unir las dos técnicas aumenta la fiabilidad del análisis y muestra relaciones entre situaciones desde dos ángulos diferentes.

Figura 12*Comparación de las técnicas Kmeans y Aglomerativo*

Nota: Elaboración propia: La figura muestra los algoritmos Aglomerativo y Kmeans, se comparan ambos para detectar similitudes y diferencias entre los grupos.

Interpretación

La figura 12 compara las dos técnicas y mediante el análisis de los clústers identificó tres tipos principales de funciones:

Escenarios administrativos, relacionadas con la gestión de usuarios y organizaciones del sistema.

Escenarios clínicos, que se refieren a la atención al paciente, la solicitud de pruebas y el registro de resultados.

Escenarios logísticos, que están ligadas al transporte, al envío de muestras y a la conclusión de tareas operativas.

La aplicación de este método mostró que la fusión de K-Means y Clustering Aglomerativo es una herramienta útil para identificar y validar estructuras funcionales en colecciones de descripciones textuales. En el ámbito de la ingeniería de requisitos, esta metodología ayudó a organizar y entender los escenarios de las tablas 1 al 12 desde un punto de vista semántico, brindando una base objetiva para la reutilización, clasificación y comparación de artefactos textuales en proyectos de software.

Capítulo 6 - Análisis

6.1 Uso con artefactos Escenarios

En la presente investigación, se implementaron varias metodologías de agrupamiento, tales como K-Means, Agrupamiento Jerárquico Aglomerativo, CLARANS y DBSCAN, sobre un conjunto de escenarios derivados de un sistema detallado en las tablas 1 al 12, abarcando tanto los títulos como los episodios narrativos. El objetivo principal fue organizar escenarios que compartiesen patrones similares en su comportamiento o propósito funcional, lo que facilitaría la clasificación, trazabilidad y reutilización de los requisitos.

Cada uno de los métodos proporcionó diferentes enfoques para la agrupación. El algoritmo HAC ofreció una visión jerárquica de cómo los escenarios se relacionan entre sí, lo que resulta útil para identificar niveles de generalización. K-Means, por su parte, presentó una clasificación más sencilla y segmentada, apta para reconocer grupos funcionales claramente definidos, como los procesos de “Gestión de usuarios”, “Procesos médicos” o “solicitud de turnos”. CLARANS, al combinar características de técnicas jerárquicas y particionales, brindó una estructura más adaptable, permitiendo la identificación de grupos intermedios entre las anteriores. Finalmente, DBSCAN facilitó la detección de escenarios que resultaron ser atípicos o poco relacionados con los demás, lo que sugiere posibles inconsistencias.

De esta manera, la aplicación de técnicas de agrupamiento a los artefactos: escenario resultó ser altamente beneficiosa para la gestión del conocimiento en el ámbito de los requisitos. Esto no solo posibilitó la agrupación de elementos similares, sino también la identificación de redundancias, la propuesta de reutilizaciones y el apoyo en la creación de modelos de requisitos que sean más coherentes y organizados.

6.2 Evaluación de modelos

La evaluación de los modelos de agrupamiento generados fue un aspecto crucial para establecer la efectividad de cada método en los contextos estudiados. Dado que no existen clasificaciones previas, lo que plantea un desafío en el aprendizaje no supervisado, se utilizaron métricas internas de validación, como la silueta media, la distancia dentro de los clústers y la separación entre clústers, además de realizar un análisis cualitativo que se fundamentó en la interpretación semántica de los escenarios formados.

En el ámbito cuantitativo, los enfoques K-Means y CLARANS demostraron una coherencia interna superior, reflejada en valores promedio de silueta más altos, sugiriendo que los escenarios en el mismo clúster presentan mayor similitud entre ellos, en comparación con aquellos de otros clústers. Por su parte, la técnica de HAC facilitó la exploración de diversos niveles de detalle, evidenciando que los agrupamientos de mayor nivel lo que significa menos clústers sugieren divisiones funcionales amplias, mientras que los niveles más bajos lo que significa más clústers descubren subgrupos específicos en términos de comportamiento. DBSCAN, por otro lado, destacó un número reducido de escenarios considerados ruidosos o atípicos, lo que resulta útil para la revisión y limpieza del conjunto de requerimientos. Desde una perspectiva cualitativa, los clústers obtenidos se analizaron según el significado de los escenarios. La evaluación evidencia que las técnicas de agrupamiento son herramientas efectivas para facilitar la toma de decisiones en la fase de análisis, ayudando a identificar patrones y conexiones entre requerimientos.

En síntesis, el análisis de los modelos permitió confirmar que emplear una variedad de algoritmos complementarios proporciona una comprensión más profunda del conjunto de escenarios, combinando precisión estructural y capacidad para detectar anomalías. Esto respalda la utilización del clustering como un método de apoyo en la ingeniería de requerimientos basada en artefactos.

Capítulo 7 - Resultados

7.1 Diseño con uso de LLM

El enfoque metodológico de esta investigación incorporó Modelos de Lenguaje de Gran Escala para facilitar el análisis semántico y el procesamiento de los escenarios. El objetivo fue mejorar las técnicas de agrupamiento al aprovechar la comprensión contextual del lenguaje natural, lo que permite una representación más precisa de las similitudes entre distintos escenarios.

Se utilizaron los LLM para generar representaciones vectoriales denominado embeddings de los textos vinculados a los títulos y episodios de los escenarios TTPS. Estas representaciones semánticas reflejan las conexiones subyacentes entre palabras, conceptos y estructuras narrativas, elevando la calidad de las métricas de similitud que utilizan los algoritmos de agrupamiento. A diferencia de comparar escenarios solo por coincidencias en el léxico, el modelo pudo discernir afinidades semánticas, como reconocer que “Registrar médico” y “Dar de alta usuario” comparten un objetivo administrativo común, a pesar de las diferencias superficiales en el lenguaje utilizado.

Luego de aplicar los embeddings producidos por el LLM, se llevó a cabo la fase de agrupamiento con los algoritmos K-Means, HAC, CLARANS y DBSCAN. Los clústers resultantes reflejaron categorías funcionales más coherentes, como la Gestión, Procesos médicos y Administración general. Además, la aplicación de los LLM facilitó la automatización de la identificación de redundancias entre los escenarios y propuso agrupaciones interpretables. Han mostrado una notable eficacia en la identificación de similitudes subyacentes entre escenarios formulados con diferentes palabras, mejorando tanto la comprensión del dominio como la reutilización sistemática de artefactos.

En conclusión, los hallazgos relacionados mediante el diseño que empleó LLM demuestran un progreso significativo en la efectividad del análisis y la categorización de situaciones en Ingeniería de Requerimientos. El uso de estos modelos ha facilitado una representación más profunda de los textos de los escenarios, logrando captar las conexiones conceptuales entre títulos, actores y objetivos descritos.

Capítulo 8 - Conclusiones

8.1 Líneas Futuras de Investigación

El estudio realizado demuestra una alta coherencia estructural en el diseño de los escenarios utilizados, lo que respalda la consistencia interna del modelo de procesos que se ha representado por dichos escenarios. Este hallazgo indica que la utilización de técnicas de Clustering facilitadas por Modelos de Lenguaje de Gran Escala (LLM) representa una herramienta eficaz para examinar tanto la estructura como la calidad conceptual de los escenarios, además de abrir nuevas posibilidades para profundizar y ampliar el uso de LLM y técnicas de agrupamiento en el ámbito de la Ingeniería de Requerimientos.

En primer lugar, sería útil aplicar este enfoque en otros dominios para evaluar su capacidad de generalización. La implementación en ámbitos diversos, como la educación o las finanzas, para verificar si la coherencia y el modularidad observadas se sostienen en escenarios con diferentes configuraciones semánticas y niveles de complejidad.

Otra línea de investigación sería establecer un modelo de Ingeniería de Requerimientos respaldado por inteligencia artificial, que se dedique a describir los procesos del sistema, también a evaluar y optimizar la calidad estructural y conceptual de los escenarios en distintos dominios.

Acrónimos

ARI: Índice de Rand Ajustado

CLARANS algorithm: Clustering Large Applications based on RANdomized Search
(Agrupamiento para Aplicaciones Grandes basado en Búsqueda Aleatoria)

DBSCAN: Density-Based Spatial Clustering of Applications with Noise
(Agrupamiento Basado en Densidad para Aplicaciones con Ruido)

HAC: Hierarchical Algorithm Clustering (Agrupamiento Jerárquico)

IA: Inteligencia Artificial

IR: Ingeniería de requerimientos

LLMs: Large Language Model (Modelos de Lenguaje a Gran Escala)

PLN: Procesamiento de Lenguaje Natural

NLP4RE: Natural Language Processing for Requirements Engineering
(Procesamiento del Lenguaje Natural para la Ingeniería de Requisitos)

RE: Requirements Engeneering (Ingeniería de requisitos)

SDLC: Software Development Life Cycle (Ciclo de vida de desarrollo de software)

Referencias

- Abdalazeim, Alaa & Meziane, Farid. (2021). A review of the generation of requirements specification in natural language using objects UML models and domain ontology. *Procedia Computer Science*. 189. 328-334.
10.1016/j.procs.2021.05.102.https://www.researchgate.net/publication/353256516_A_review_of_the_generation_of_requirements_specification_in_natural_language_using_objects_UML_models_and_domain_ontology
- Azzahra, Adelia & Nabila, Nasywa & Idhom, Mohammad & Trimono, Trimono. (2025). Clustering of the Air Pollution Standard Index (ISPU) in the Province of DKI Jakarta Using the CLARANS Algorithm. *Journal of Applied Informatics and Computing*. 9. 1219-1226.
10.30871/jaic.v9i4.9783.
https://www.researchgate.net/publication/395222487_Clustering_of_the_Air_Pollution_Standard_Index_ISPU_in_the_Province_of_DKI_Jakarta_Using_the_CLARANS_Algorithm
- Arutyunova, Anna & Röglin, Heiko. (2025). The Price of Hierarchical Clustering. *Algorithmica*. 87. 1420-1452. 10.1007/s00453-025-01327-7.
https://www.researchgate.net/publication/393322441_The_Price_of_Hierarchical_Clustering
- Bernanda, Devi Yurisca; Jawawi, Dayang N.A.; Abd Halim, Shahliza; and Adikara, Fransiskus (2024)"Natural Language Processing For Requirement Elicitation In University Using Kmeans And Meanshift Algorithm," *Baghdad Science Journal*: Vol. 21: Iss. 2, Article 29. DOI: <https://doi.org/10.21123/bsj.2024.9675>

Celi Párraga, R. J., Boné Andrade, M. F., Mora Olivero, A. P., Sarmiento Saavedra J. C. (2023). Ingeniería del Software I: Requerimientos y Modelado del Software. Editorial Grupo AEA <https://www.editorialgrupo-aea.com/index.php/EditorialGrupoAEA/catalog/view/21/27/114>

Chacón, J.E., Rastrojo, A.I. Minimum adjusted Rand index for two clusterings of a given size. *Adv Data Anal Classif* 17, 125–133 (2023).
<https://doi.org/10.1007/s11634-022-00491-w>
<https://link.springer.com/article/10.1007/s11634-022-00491-w>

Cibulková, Jana & Kupková, Barbora. (2022). Review of Visualization Methods for Categorical Data in Cluster Analysis. *Statistika: Statistics and Economy Journal*. 102. 396-408. 10.54694/stat.2022.4.
https://www.researchgate.net/publication/366353960_Review_of_Visualization_Methods_for_Categorical_Data_in_Cluster_Analysis

Eroglu, Derya Ipek & Iyigun, Cem. (2025). A Foundational Perspective for Partitional Clustering on Networks.
https://www.researchgate.net/publication/390929150_A_Foundational_Perspective_for_Partitional_Clustering_on_Networks

Eyal-Salman, Hamzeh & Hammad, Mustafa & Seriai, Abdelhak & Al-Sbou, Ahed. (2018). Semantic Clustering of Functional Requirements Using Agglomerative Hierarchical Clustering. *Information*. 9. 222.
 10.3390/info9090222.https://www.researchgate.net/publication/327410624_Semantic_Clustering_of_Functional_Requirements_Using_Agglomerative_Hierarchical_Clustering

- Ezukwoke, Kenneth & Zareian, Samaneh. (2019). KMEANS AND KERNEL KMEANS A comparative study of classical and kernel kmeans for data clustering. 10.13140/RG.2.2.29297.43361.
https://www.researchgate.net/publication/337932249_KMEANS_AND_KERNEL_KMEANS_A_comparative_study_of_classical_and_kernel_kmeans_for_data_clustering
- Ferrari, A., Gnesi, S., Tolomei, G. (2013). Using Clustering to Improve the Structure of Natural Language Requirements Documents. In: Doerr, J., Opdahl, A.L. (eds) Requirements Engineering: Foundation for Software Quality. REFSQ 2013. Lecture Notes in Computer Science, vol 7830. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37422-7_3
- Glinz M (2020) Standard glossary for the certified professional for requirements engineering (CPRE) studies and exam v2.0.0. Technical report, International Requirements Engineering Board e.V., October 2020
https://expleoacademy.com/dach/wp-content/uploads/sites/4/2020/11/ireb_cpre_glossary_17.pdf
- Jin, X., Han, J. (2011). Partitional Clustering. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.
https://doi.org/10.1007/978-0-387-30164-8_631
- Jowhari, Hossein. (2024). Hierarchical Clustering via Local Search. 10.48550/arXiv.2405.15983.
https://www.researchgate.net/publication/380907537_Hierarchical_Clustering_via_Local_Search

- Latif, Maria & Aslam, Toseef & Sehar, Palwashay. (2018). Impact of domain knowledge in phase of requirement engineering. *International Journal of Advanced Computer Research*. 3. 54.
https://www.researchgate.net/publication/335777274_Impact_of_domain_knowledge_in_phase_of_requirement_engineering
- Lim, S., Henriksson, A. & Zdravkovic, J. Data-Driven Requirements Elicitation: A Systematic Literature Review. *SN COMPUT. SCI.* **2**, 16 (2021).
<https://doi.org/10.1007/s42979-020-00416-4>
- Lu, H., Uddin, S. Unsupervised machine learning for disease prediction: a comparative performance analysis using multiple datasets. *Health Technol.* **14**, 141–154 (2024). <https://doi.org/10.1007/s12553-023-00805-8>
- Mazak-Huemer, Alexandra & Huemer, Christian & Vierhauser, Michael & Janger, Jürgen. (2025). Requirements Engineering for a Web-based Research, Technology & Innovation Monitoring Tool. 10.48550/arXiv.2501.10872.
<https://publications.jrc.ec.europa.eu/repository/handle/JRC139508>
- Murtagh, Fionn & Legendre, Pierre. (2011). Ward's Hierarchical Clustering Method: Clustering Criterion and Agglomerative Algorithm. [\(PDF\) Ward's Hierarchical Clustering Method: Clustering Criterion and Agglomerative Algorithm](#)
- Necula, S.-C.; Dumitriu, F.; Greavu-Şerban, V. A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering. *Electronics* **2024**, *13*, 2055.
<https://doi.org/10.3390/electronics13112055>

Pérez, G., Mostaccio, C., & Antonelli, L. (2024). Evaluación de modelos de procesamiento de lenguaje natural para medir similaridad entre escenarios escritos en español. In *WER*.http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER24/WER2024_paper_4.pdf

Pressman, R. (2010). *Ingeniería del software: Un enfoque práctico* (7.^a ed.). McGraw-Hill.

https://www.javier8a.com/umb/analisis/Libro_Pressman_7.pdf

Schleifer, Simon & Lungu, Adriana & Kruse, Benjamin & Putten, Sebastiaan & Goetz, Stefan & Wartzack, Sandro. (2025). Use case identification of natural language system requirements with graph-based clustering. *Design Science*. 11. 10.1017/dsj.2025.10019.

https://www.researchgate.net/publication/393874992_Use_case_identification_of_natural_language_system_requirements_with_graph-based_clustering

Z. Li, Y. Li, W. Lu and J. Huang, "Crowdsourcing Logistics Pricing Optimization Model Based on DBSCAN Clustering Algorithm," in *IEEE Access*, vol. 8, pp. 92615-92626, 2020, doi: 10.1109/ACCESS.2020.2995063.

<https://ieeexplore.ieee.org/document/9094209>

Zhao, Jinyu & Hao, Yi & Rashtchian, Cyrus. (2020). Unsupervised Embedding of Hierarchical Structure in Euclidean Space.

10.48550/arXiv.2010.16055.<https://arxiv.org/pdf/2010.16055>

Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M., Chioasca, E.-V., & Batista-Navarro, R. T. (2021). Natural Language Processing for Requirements Engineering: A Systematic Mapping Study. *ACM*

Computing Surveys, 54(3), 1–41. Article 55.

<https://doi.org/10.1145/3444689>

Zhao, Ying & Karypis, George. (2002). Comparison of Agglomerative and Partitional Document Clustering Algorithms. 13.

https://www.researchgate.net/publication/235207548_Comparison_of_Agglomerative_and_Partitional_Document_Clustering_Algorithms