Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Trabajo Fin de Grado

Diseño y Desarrollo de un Sistema de Respuesta a Ciberamenazas Basado en Large Language Models y Retrieval Augmented Generation

Rodrigo Tavares de Pina Simões

20 de junio de 2025

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Trabajo Fin de Grado

Título: Diseño y Desarrollo de un Sistema de Respuesta a Ciberamenazas Basado en Large Language Models y Retrieval Augmented Generation

Autor: D. Rodrigo Tavares de Pina Simões

Tutor: D. Luis Pérez Miguel

Ponente: D. Xavier Larriva-Novo

Departamento: Departamento de Ingeniería de Sistemas Telemáticos

Miembros del Tribunal

Presidente: D. . . . . .

Vocal: D. . . . . ..

Secretario: D. . . . . ..

Suplente: D. . . . . ..

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de: . . . . . .

Madrid, a de de 20...

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Trabajo Fin de Grado

Diseño y Desarrollo de un Sistema de Respuesta a Ciberamenazas Basado en Large Language Models y Retrieval Augmented Generation

Rodrigo Tavares de Pina Simões

20 de junio de 2025

Resumen

Hoy en día, la ciberseguridad se ha convertido en un pilar esencial para empresas y organizaciones, impulsada por el aumento constante de ciberataques y su creciente sofisticación. Uno de los principales retos consiste en detectar y responder de forma ágil a estas amenazas, tanto si provienen del exterior como si se manifiestan a través de comportamientos anómalos dentro de los propios sistemas corporativos.

Para examinar y reaccionar eficazmente ante estas amenazas, los sistemas de *Gestión de Información y Eventos de Seguridad* se han vuelto indispensables. Estos sistemas recopilan y vinculan registros (logs) de distintas fuentes, facilitando una respuesta rápida y eficaz ante posibles incidentes. No obstante, examinar todos estos logs puede ser una tarea que consume una gran cantidad de tiempo y recursos.

En este contexto, la integración de modelos de *Inteligencia Artificial Generativa*, a través de un enfoque de *Generación Mejorada por Recuperación de Información*, mejora el proceso de evaluación y análisis. Esta metodología no solo facilita el estudio de los eventos registrados, sino que también potencia el análisis con información externa. De esta forma, se optimiza la exactitud de las sugerencias y se disminuye el tiempo necesario para reaccionar ante los incidentes. Al incluir información adicional que contextualiza los patrones de ataque, los *Modelos de Lenguaje Avanzados* pueden identificar con más precisión las vulnerabilidades y secuencias de acciones maliciosas, lo que simplifica la puesta en marcha de estrategias de mitigación rápidas y eficaces.

Este proyecto propone la creación de un sistema que utilice la *Inteligencia Artificial Generativa* y el enfoque de *Generación Mejorada Por Recuperación* para analizar los registros que provienen de un sistema de *Gestión de Información y Eventos de Seguridad*. Estos logs serán procesados por un modelo de lenguaje que identificará y clasificará las amenazas en función de su nivel de riesgo. A partir de este análisis, el sistema proporcionará al usuario sugerencias útiles y *Cursos de Acción*, para ayudar a mitigar las consecuencias de cada amenaza, acelerando la reacción y mejorando la eficiencia de los equipos de ciberseguridad.

El objetivo es validar la efectividad de este enfoque de *Generación Mejorada por Recuperación de Información* en la detección de intrusiones y en la generación de recomendaciones precisas en un periodo de tiempo reducido. Para ello, se plantean dos escenarios de validación: el primero consistirá en un ataque simulado utilizando la herramienta CALDERA, y el segundo se basará en el análisis de los logs derivados de una práctica de tipo *Capture the Flag*. En ambos casos, se evaluará no solo la capacidad del sistema para identificar las amenazas, sino también la calidad y utilidad del informe generado, incluyendo la relevancia de las recomendaciones propuestas.

Summary

Nowadays, cybersecurity has become a fundamental pillar for companies and organiza-

tions, driven by the constant rise in cyberattacks and their increasing sophistication. One of the main challenges lies in detecting and responding quickly to these threats, whether they originate externally or manifest through anomalous behavior within corporate systems.

To effectively examine and respond to such threats, *Security Information and Event Management* systems have become essential. These systems collect and correlate *logs* from various sources, enabling a rapid and efficient response to potential incidents. However, reviewing all these logs can be a time-consuming and resource-intensive task.

In this context, the integration of *Generative Artificial Intelligence* models through a *Retrieval-Augmented Generation* approach enhances the evaluation and analysis process. This methodology not only facilitates the study of recorded events but also strengthens the analysis with external information. As a result, the accuracy of suggestions is improved and the time needed to respond to incidents is reduced. By incorporating additional information that contextualizes attack patterns, *Large Language Models* can more accurately identify vulnerabilities and sequences of malicious actions, simplifying the implementation of quick and effective mitigation strategies.

This project proposes the development of a system that employs generative artificial intelligence and the *Retrieval-Augmented Generation* approach to analyze logs generated by a *Security Information and Event Management* system. These logs will be processed by a language model that will identify and classify threats according to their level of risk. Based on this analysis, the system will provide the user with useful suggestions and *Courses of Action* to help mitigate the consequences of each threat, accelerating response and improving the efficiency of cybersecurity teams.

The aim is to validate the effectiveness of the *Retrieval-Augmented Generation* approach in intrusion detection and in generating accurate recommendations within a short time frame. To this end, two validation scenarios are proposed: the first involves a simulated attack using the CALDERA tool, and the second is based on the analysis of logs generated during a *Capture the Flag* exercise. In both cases, the system will be evaluated not only on its ability to identify threats, but also on the quality and usefulness of the generated report, including the relevance of the proposed recommendations.

Palabras Clave

RAG, LLM, IA, Ciberseguridad, Ciberdefensa, SIEM, Cursos de Acción (COAs), Logs, Automatización

Keywords

RAG LLM, AI, Cibersecurity, Ciberdefense, SIEM, Courses of Action (COAs), Logs, Automatization

Índice

Índice de figuras

Índice de tablas

Lista de acrónimos

IA / AI *Inteligencia Artificial / Artificial Inteligence*

RAG *Retrieval-Augmented Generation (Generación Aumentada por Recuperación)*

LLM *Large Language model (Modelo de Lenguaje a Gran Escala)*

SIEM *Security Information and Event Management system (sistema de Gestión de Información y Eventos de Seguridad)*

COA *Course of Action (Curso de Acción)*

NLP *Natural Language Processing (Procesamiento de Lenguaje Natural)*

CoT *Chain of Thought (Cadena de Pensamiento)*

IoT *Internet of Things (Internet de las Cosas)*

IDS *Intrusion Detection Systems (Sistemas de Detección de Intrusiones)*

SOC *Security Operations Center (Centros de Operaciones de Seguridad)*

TTP *Tácticas, Técnicas y Procedimientos*

1. Introducción y objetivos

1.1. Introducción

En la era en la que vivimos, los datos son un recurso estratégico esencial. Su crecimiento exponencial, impulsado por la digitalización, ha transformado la forma en que trabajamos, nos comunicamos y tomamos decisiones. Esta evolución, comparable a la revolución que supuso Internet, nos ha llevado de un mundo donde buscar información requería tiempo y esfuerzo físico a un presente donde basta un clic para acceder a millones de resultados en segundos. Sin embargo, este acceso masivo plantea un nuevo desafío: ¿cómo gestionamos e interpretamos tal cantidad de información sin perdernos en el ruido?

Este problema se vuelve aún más crítico en el ámbito de la ciberseguridad. En un entorno digitalizado y conectado, cada interacción, transacción y proceso genera registros o logs que deben analizarse para detectar posibles amenazas. Diversas herramientas como los EDR (Endpoint Detection and Response), IDS (Intrusion Detection Systems) y SIEM (Security Information and Event Management) son clave para gestionar estos eventos. Estas herramientas vigilan el tráfico y el comportamiento de los sistemas, permitiendo correlacionar eventos para identificar patrones de amenaza. En este trabajo nos centraremos en los sistemas SIEM, en concreto Wazuh [1], que integran múltiples fuentes de eventos, proporcionando una visión centralizada para su análisis. Sin embargo, ante la creciente complejidad de los ataques y el volumen de datos a procesar, se hace evidente que estas soluciones necesitan apoyarse en tecnologías más avanzadas para mantener su eficacia.

Además de estas plataformas, otro recurso esencial en la lucha contra las amenazas son las contramedidas normativas, como las regulaciones específicas, los estándares de ciberseguridad y los marcos de modelado de amenazas. Entre estos, destaca MITRE ATT&CK [2], que proporciona un marco estructurado para identificar y analizar patrones de ataque a través de Tácticas, Técnicas y Procedimientos (TTPs) utilizados por los adversarios. Aunque la matriz ATT&CK también incluye información sobre mitigaciones, su enfoque principal es describir el comportamiento de los atacantes en lugar de definir contramedidas directas.

Todos estos elementos, si bien facilitan la toma de decisiones, incrementan también la carga cognitiva de los analistas, que deben considerar un volumen de información cada vez mayor. Es aquí donde los Modelos de Lenguaje Avanzados (Large Language Models, LLMs) y las metodologías de Generación Aumentada por Recuperación (Retrieval Augmented Generation, RAG) pueden marcar la diferencia.

Los LLMs, una rama de la Inteligencia Artificial Generativa, destacan por su capacidad de procesar grandes volúmenes de datos no estructurados y generar texto coherente, permitiendo una interacción más natural entre humanos y sistemas. Sin embargo, su ge-

neralidad y la posibilidad de alucinaciones (generación de información incorrecta) pueden limitar su aplicación directa en contextos críticos como la ciberseguridad. Al combinar estas capacidades con metodologías RAG, que permiten consultar información estructurada y contextual de fuentes fiables, es posible mitigar estos riesgos y mejorar tanto la precisión como la relevancia de las respuestas.

Este Trabajo de Fin de Grado se centra en explorar esta integración, utilizando herramientas como Wazuh para gestionar logs y CALDERA para simular ataques realistas, con el objetivo de diseñar un sistema que clasifique las amenazas según su criticidad y proponga Cursos de Acción (COAs) efectivos. Al mismo tiempo, se busca apoyar este análisis en marcos y catálogos como MITRE ATT&CK, que proporcionan una clasificación detallada de TTPs utilizados por los atacantes. Esto permitirá al sistema ofrecer recomendaciones fundamentadas y en línea con las mejores prácticas de la industria.

La complejidad del análisis en ciberseguridad radica no solo en identificar posibles amenazas, sino en priorizarlas correctamente para evitar la parálisis por exceso de información. Este proyecto busca dar respuesta a este reto, creando un sistema capaz de filtrar y analizar eventos con agilidad. A través de la combinación de un SIEM con un modelo de lenguaje avanzado apoyado en RAG, se pretende ofrecer no solo detección, sino una verdadera capacidad de respuesta basada en datos reales y contextuales.

Con el apoyo de CALDERA [3] y de la práctica de Capture the Flag [4] de la asignatura SEGU de esta escuela, se replicarán escenarios de intrusión que generen logs representativos, permitiendo evaluar cómo el sistema responde a diferentes niveles de amenaza. Así, no solo pondremos a prueba la precisión del análisis, sino también la viabilidad de las soluciones propuestas para mitigar riesgos en entornos reales.

En definitiva, este proyecto no solo busca mejorar las capacidades técnicas de los sistemas de detección de intrusiones, sino también aportar un enfoque innovador a la ciberdefensa, donde la tecnología no se limita a reaccionar, sino que anticipa, interpreta y actúa con precisión frente a las amenazas del entorno digital.

1.2. Objetivos

El objetivo principal de este TFG es desarrollar un sistema que integre modelos de IA Generativa y técnicas RAG dentro de un entorno de ciberseguridad. Este sistema estará diseñado para analizar y priorizar amenazas detectadas en registros (logs) generados por un SIEM (Wazuh), en función de los distintos escenarios de prueba, y propondrá soluciones concretas basadas en datos estructurados y experiencias previas.

Para alcanzar este propósito, se han definido los siguientes objetivos específicos:

- Comprender el funcionamiento del modelo RAG y determinar el mejor método para implementarlo en los casos estudiados.
- Diseñar un flujo de integración entre los registros generados por Wazuh y la herramienta de simulación CALDERA, asegurando una compatibilidad completa entre ambos sistemas.
- Implementar un modelo de IA generativa apoyado en RAG que utilice datos contextuales relevantes (catálogos de amenazas, estándares como MITRE ATT&CK, y casos históricos) para analizar y priorizar amenazas.

- Proponer cursos de acción (COAs) personalizados para mitigar las amenazas detectadas, basándose en el análisis contextual realizado por el LLM teniendo en cuenta las bases de datos asociadas, los catálogos de mitigaciones y las regulaciones de seguridad pertinentes.
- Evaluar y validar la eficacia del sistema para clasificar amenazas según su nivel de gravedad, priorizando las alertas más relevantes y midiendo su capacidad para reducir el tiempo de respuesta frente a intrusiones simuladas, mejorando la precisión en la identificación y priorización de amenazas.

Con estos objetivos, este proyecto no solo busca una implementación técnica, sino también una contribución real al campo de la ciberseguridad, mejorando las capacidades de análisis y respuesta en un entorno cada vez más complejo y dinámico.

2. Marco Teórico

En este capítulo se presentan los fundamentos conceptuales y tecnológicos que sustentan el desarrollo del sistema propuesto. Se abordan las principales tecnologías involucradas, comenzando por los LLMs, que constituyen la base del procesamiento y análisis automatizado de texto. A continuación, se profundiza en el enfoque RAG, una metodología que permite superar las limitaciones de los LLMs tradicionales al integrar información contextual proveniente de fuentes externas. Finalmente, se analiza el papel de los sistemas SIEM, con énfasis en Wazuh, y su integración con modelos RAG para optimizar la detección, clasificación y mitigación de amenazas en el ámbito de la ciberseguridad.

2.1. LLMs

Los Modelos de Lenguaje a Gran Escala han transformado el panorama tecnológico, aportando soluciones avanzadas en múltiples disciplinas, incluida la ciberseguridad. Estos modelos, han demostrado capacidades sobresalientes en Comprensión y Generación de Lenguaje Natural (Natural Language Processing, NLP), lo que les permite abordar tareas complejas con un alto grado de eficiencia [5, 6, 7]. En el ámbito de la ciberseguridad, los LLMs están redefiniendo las estrategias para detectar, mitigar y prevenir amenazas digitales, al tiempo que generan desafíos éticos y técnicos significativos [8, 9, 10].

Los LLMs han mostrado una versatilidad sin precedentes en NLP, gracias a su preentrenamiento en conjuntos de datos masivos y su capacidad para comprender contextos complejos. Esto les permite ser adaptados a tareas específicas mediante procesos de "fine-tuning", optimizando su desempeño para resolver problemas concretos en ciberseguridad, como la detección de anomalías y vulnerabilidades [11, 12] o la generación de informes de inteligencia de amenazas [13]. A pesar de estas capacidades, los LLMs enfrentan limitaciones importantes. Una de las más críticas es el fenómeno conocido como "alucinación"[14], que ocurre cuando generan respuestas incorrectas o no fundamentadas en hechos reales. Esto puede comprometer su fiabilidad en tareas sensibles, como la clasificación de amenazas o la identificación de vulnerabilidades. Para abordar este problema, se han propuesto varias soluciones que buscan mejorar la precisión de estos modelos:

- Integración de RAG: Este enfoque permite a los modelos consultar bases de datos externas en tiempo real, mejorando la precisión de las respuestas. Este método es especialmente relevante para escenarios que requieren precisión [15], como la detección de amenazas en logs o la clasificación de vulnerabilidades.
- Optimización de prompts: El diseño adecuado de las indicaciones que se dan al modelo (prompts) es clave para obtener mejores resultados. Estudios recientes, como el de Qi et al. [16], han demostrado que los formatos avanzados, como los prompts de cadena de pensamiento (Chain of Thought, CoT), pueden reducir errores. Sin

embargo, estos métodos aún enfrentan desafíos, como las altas tasas de falsos positivos.

- Mecanismos de autocorrección: Iniciativas como SELFCHECKGPT [17] utilizan estrategias de autoevaluación y aprendizaje contrastivo para minimizar las respuestas inexactas, algo que podría implementarse en el análisis de ciberseguridad.

Otra limitación significativa es la falta de transparencia en algunos modelos propietarios, como ChatGPT [18] o Gemini [19]. Aunque estos modelos ofrecen un alto rendimiento, no permiten personalización ni auditorías independientes, lo que limita su aplicabilidad en contextos críticos. Por otro lado, los modelos de código abierto, como Llama [20] o Mixtral [21], proporcionan mayor flexibilidad al permitir ajustes específicos, aunque suelen estar por detrás en términos de precisión y escalabilidad.

Además de estos factores, el costo computacional de entrenar y operar LLMs plantea desafíos de sostenibilidad [22], tanto desde el punto de vista económico como ambiental. Esto, junto con la propagación de desinformación y parcialidad [23], ha impulsado la investigación en métodos más eficientes de entrenamiento y en arquitecturas que reduzcan el consumo energético de estos sistemas [24]. Esta dimensión de las LLMs es explorada en más detalle en el *Anexo A: Aspectos éticos, económicos, sociales y ambientales*.

Por último, otro desafío clave es la falta de explicabilidad de los LLMs, que a menudo son tratados como "cajas negras"[25]. En el ámbito de la ciberseguridad, donde la transparencia es esencial, esto puede ser problemático. Para superar esta barrera, se están explorando distintas líneas de investigación:

- Modelos explicables: La creación de modelos que proporcionen justificaciones verificables para sus decisiones es crucial. Por ejemplo, herramientas como ChatIDS [26], que explican alertas de sistemas IDS mediante LLMs, pueden ser un punto de partida para desarrollos más avanzados en esta área.
- Interfaces interactivas para no expertos [27]: Iniciativas para mejorar la accesibilidad de la información a usuarios no técnicos pueden extenderse a sistemas que permitan consultas interactivas y aclaraciones adicionales.

A pesar de todos estos desafíos, los LLMs continúan consolidándose como herramientas clave en ciberseguridad. Su capacidad para procesar grandes volúmenes de datos, identificar patrones complejos y generar soluciones automatizadas los convierte en aliados fundamentales para abordar amenazas emergentes.

2.2. Enfoque RAG

El enfoque de la Generación Aumentada por Recuperación representa una evolución significativa en el uso de LLMs, ya que afronta retos importantes para garantizar su efectividad en áreas especializadas. RAG combina la capacidad generativa de los LLMs con técnicas avanzadas de recuperación de información para generar respuestas contextualmente relevantes. Este método ha demostrado ser especialmente útil para superar limitaciones como las alucinaciones, mencionadas anteriormente, y en escenarios donde los datos externos son dinámicos, especializados o sensibles al tiempo, como en la ciberseguridad.

En resumen, el sistema RAG consta de dos bloques principales, representados en la Figura 2.1:

- Módulo de Recuperación (Retriever): Su función es identificar y recuperar documentos relevantes de una base de conocimientos, reduciendo el conjunto de documentos inicial a un subconjunto de tamaño manejable, el cual contenga información suficiente para respaldar el razonamiento y la generación de una respuesta precisa a la consulta (query). Para lograrlo, se transforma tanto la consulta como los documentos potenciales en representaciones vectoriales (embeddings) mediante redes neuronales. A partir de estas representaciones, se calcula la similitud entre los vectores de la consulta y los documentos. Esto se consigue mediante métricas como el producto punto/escalar, priorizando aquellos documentos con mayor relevancia para ser procesados en el módulo generativo.
- Módulo Generativo (Generator): Este componente, típicamente implementado mediante un LLM, se encarga de sintetizar respuestas coherentes y precisas. Los modelos generativos operan prediciendo la distribución de probabilidad del siguiente token, dada una secuencia previa de tokens. En el contexto de RAG, el modelo toma como entrada la query y los documentos recuperados por el retriever, generando una respuesta mediante la predicción secuencial de tokens. Este enfoque dinámico permite que el LLM produzca respuestas basadas en información estática (la query) y contextual (los documentos recuperados), maximizando la efectividad del sistema en la tarea de generación de respuestas.

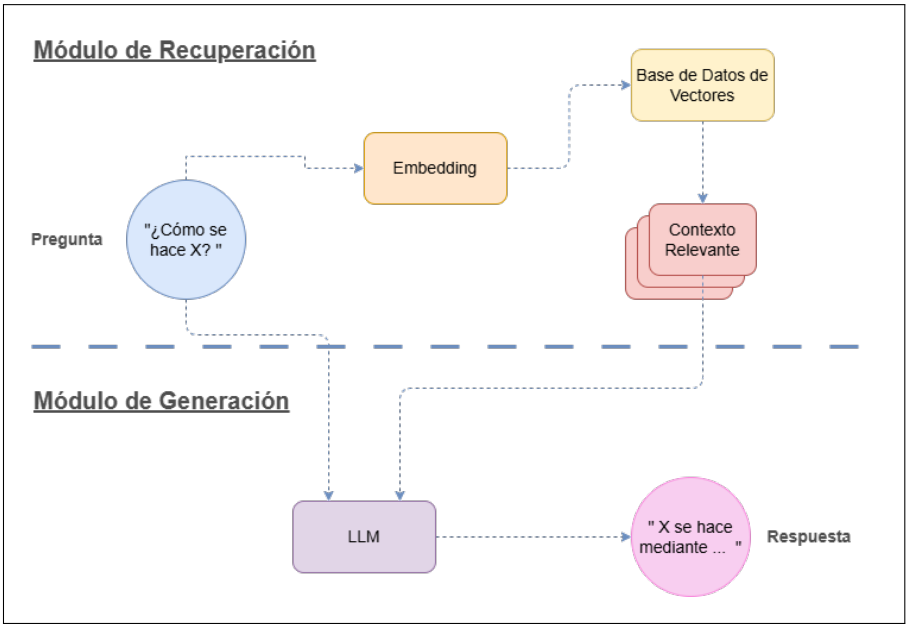

Figura 2.1: Arquitectura general de un sistema RAG.
Imagen adaptada de [28].

2.2.1. Proceso del sistema RAG

El enfoque RAG sigue un proceso bien definido que combina la recuperación de fragmentos relevantes de una base de conocimiento con la generación de respuestas precisas y contextualizadas. Este "pipeline" se compone de cuatro etapas principales: preprocesamiento de datos, consulta del usuario, recuperación de información y generación de respuestas.

1. Preprocesamiento de Datos

   El primer paso en un sistema RAG consiste en preparar la base de conocimiento que servirá como fuente para la recuperación de información (Figura 2.2). Los documentos se dividen en fragmentos o "splits" manejables, que pueden ser secciones, párrafos o bloques de texto. Este proceso, conocido como segmentación, facilita la indexación y recuperación posteriores.

   Posteriormente, cada fragmento se convierte en una representación vectorial mediante técnicas de incrustación semántica (embedding), utilizando tanto modelos de aprendizaje profundo, como BERT [29], o LLMs con su propio modelo de embedding, como GPT [30]. Estos vectores capturan el significado intrínseco del texto, permitiendo búsquedas basadas en similitud semántica en lugar de coincidencias literales.

   Una vez creados los embeddings, los fragmentos se almacenan en sistemas de indexación como FAISS [31], Elasticsearch [32] o Chroma [33]. Estas tecnologías aseguran una recuperación rápida y eficiente de los fragmentos más relevantes durante la fase de consulta.

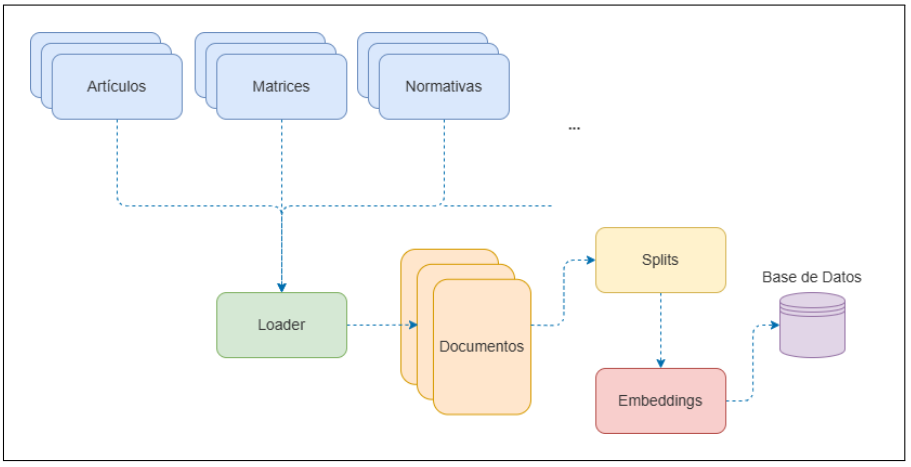

Figura 2.2: Preprocesamiento de Datos.
Imagen adaptada de [28].

2. Consulta del Usuario

   Cuando un usuario formula una consulta, esta se procesa para convertirla en un vector utilizando el mismo modelo de embedding aplicado al preprocesamiento. Este vector encapsula la semántica de la consulta, permitiendo que el sistema busque fragmentos relevantes no solo por palabras clave, sino también por su significado implícito.

3. Recuperación de Información (Retrieval)

   En esta etapa, el sistema compara el vector de la consulta con los vectores previamente indexados. Algoritmos [34] como *k-NN* (k-Nearest Neighbors) o *ANN* (Approximate Nearest Neighbors) identifican los fragmentos más cercanos en términos de similitud semántica. Posteriormente, el sistema devuelve los fragmentos más relevantes, que servirán como base para la generación de respuestas.

Para llevar a cabo esta comparación, es necesario definir una métrica que cuantifique la similitud (o distancia) entre vectores. Las métricas más comúnmente empleadas en este tipo de sistemas son:

- **Similitud del coseno (cosine similarity):** Mide el ángulo entre dos vectores, evaluando así su orientación independientemente de su magnitud. Es ampliamente utilizada en tareas de procesamiento de lenguaje natural [35], ya que captura la similitud semántica de manera efectiva.

$$\text{cosine}(x, y) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|}$$

- **Distancia euclídea (L2):** Calcula la raíz cuadrada de la suma de las diferencias al cuadrado entre las componentes de los vectores. Es adecuada para espacios donde la magnitud de los vectores es relevante, aunque puede no ser óptima en contextos semánticos.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- **Producto interno (dot product):** Evalúa la proyección de un vector sobre otro, resultando en un valor elevado cuando los vectores apuntan en la misma dirección. Aunque similar a la similitud del coseno, no normaliza los vectores, por lo que la magnitud afecta el resultado.

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$$

- **Distancia Manhattan (L1):** Suma las diferencias absolutas entre cada dimensión de los vectores. Es menos sensible a valores extremos y puede ser útil en ciertos espacios discretos, aunque es menos común en sistemas de embeddings textuales.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

4. Generación de Respuestas Contextuales (Generation)

El modelo generativo utiliza una combinación de los fragmentos recuperados y la consulta original para construir una respuesta coherente y precisa. Los fragmentos seleccionados se consolidan en un bloque único de texto, asegurando que este respete los límites de tokens del modelo. Posteriormente, se elabora un prompt que combina la consulta original y el contexto relevante, estructurado en secciones como "Contexto" y "Pregunta". Este prompt optimizado se introduce en el modelo generativo, que produce una respuesta adaptada a las necesidades del usuario. Además, parámetros como la temperatura del modelo pueden ajustarse para controlar el grado de creatividad o precisión en las respuestas.

2.2.2. Ventajas del Enfoque RAG frente a Sistemas Tradicionales de LLM

El enfoque de Generación Aumkentada por Recuperación supone un avance significativo frente a los sistemas que dependen exclusivamente de un modelo de lenguaje (LLM) "fine-tuneado". Tradicionalmente, el entrenamiento específico (fine-tuning) de un modelo generativo ha sido la estrategia principal para mejorar su rendimiento en dominios concretos. Sin embargo, esta metodología presenta limitaciones importantes, como el alto coste computacional, la necesidad de datos etiquetados y la pérdida de flexibilidad al enfrentarse a información dinámica o en constante cambio. En este contexto, RAG aporta ventajas clave que hacen que sea una solución más eficaz y escalable.

Una de las mejoras más relevantes es la mitigación de las "alucinaciones"generadas por los LLMs. Estos modelos suelen producir respuestas plausibles pero incorrectas cuando carecen del contexto necesario, lo que puede resultar problemático en áreas críticas como la ciberseguridad. Con RAG, el modelo recupera información directamente de bases de conocimiento externas antes de generar una respuesta. Este enfoque no solo reduce el riesgo de errores, sino que también aumenta la confianza en las respuestas al estar respaldadas por datos específicos.

Además, RAG destaca por su adaptabilidad a dominios especializados. Mientras que un modelo "fine-tuneado" requiere reentrenamiento para cada nueva área de aplicación, RAG permite integrar bases de conocimiento dinámicas y actualizadas sin modificar el modelo generativo. Esto resulta especialmente útil en contextos como la ciberseguridad, donde los datos evolucionan rápidamente y los ataques adoptan constantemente nuevas tácticas. En lugar de gastar recursos en reentrenamientos frecuentes, RAG garantiza que las respuestas se ajusten al contexto actual, optimizando tanto el tiempo como los costes.

Otro beneficio clave de RAG es su capacidad para mantener las respuestas actualizadas. A diferencia de los modelos entrenados en un conjunto de datos estático, RAG trabaja con bases de conocimiento dinámicas que se actualizan en tiempo real. Esto lo convierte en una solución ideal para sistemas que requieren información siempre relevante, como la detección de amenazas emergentes. Este enfoque elimina la necesidad de realizar actualizaciones periódicas al modelo base mediante fine-tuning, permitiendo una mayor flexibilidad operativa.

En términos de eficiencia operativa, los sistemas RAG también destacan. Al delegar parte del trabajo al sistema de recuperación, el modelo generativo puede centrarse únicamente en sintetizar la información recuperada, lo que reduce la carga computacional y mejora la escalabilidad en entornos con grandes volúmenes de consultas.

2.2.3. Limitaciones del Enfoque RAG y Soluciones Potenciales

Aunque RAG presenta numerosas ventajas, también enfrenta retos específicos. Uno de los problemas más notables es la latencia, ya que el proceso de recuperación de información y generación puede ser más lento que el de un modelo LLM estándar. Para abordar este inconveniente, se están explorando optimizaciones como el uso de sistemas de indexación eficientes (e.g., FAISS, Annoy [36]) y técnicas de priorización que seleccionen únicamente los fragmentos más relevantes.

Otra limitación es la dependencia de la calidad de los datos. Si las bases de conocimiento contienen información incorrecta o incompleta, las respuestas generadas reflejarán esos

errores. Esto subraya la importancia de contar con bases de datos validadas y fuentes fiables, especialmente en dominios sensibles como la ciberseguridad. En este trabajo, se utilizarán documentos certificados provenientes de catálogos reconocidos como NIST o MITRE ATT&CK [2] y regulaciones de ciberseguridad, de forma que se minimize este riesgo.

El límite de tokens de los modelos generativos también representa un desafío para este tipo de sistemas, ya que solo pueden procesar un número finito de palabras en cada consulta. Este problema se aborda mediante técnicas de compresión de texto y selección inteligente de contextos, lo que permite al sistema priorizar la información más relevante.

Por último, la seguridad y privacidad de los datos es un aspecto crítico al implementar RAG en áreas como ciberseguridad. La integración de sistemas robustos de anonimización y cifrado, junto con recuperadores locales en lugar de sistemas basados en la nube, se plantea como una solución viable.

2.2.4. Futuras Direcciones en el Desarrollo de RAG

El futuro del enfoque RAG apunta a superar estas limitaciones mediante líneas de investigación prometedoras. Una de ellas es la recuperación multimodal [28], que busca integrar datos de diferentes formatos, como texto, imágenes y bases de datos estructuradas, para enriquecer la calidad de las respuestas. Esto podría ser especialmente útil en dominios complejos donde la información relevante no está limitada a un solo tipo de fuente.

Otra dirección importante es la incorporación de grafos de conocimiento, como LangGraph [37], que permiten representar relaciones complejas entre entidades y mejorar la precisión en la recuperación de información. Estos grafos ofrecen una representación semántica más rica, especialmente valiosa en tareas de análisis de amenazas en ciberseguridad.

Además, se está explorando el desarrollo de modelos generativos más eficientes, capaces de procesar mayores cantidades de texto o realizar múltiples iteraciones de recuperación, mejorando así la profundidad y precisión de las respuestas.

Finalmente, el aprendizaje continuo también emerge como una solución prometedora para garantizar que las bases de conocimiento y los modelos generativos mantengan su relevancia a medida que evoluciona el contexto operativo. Esto permitiría actualizar el sistema de manera incremental sin necesidad de realizar entrenamientos completos, manteniendo un rendimiento óptimo frente a cambios constantes en los datos.

En resumen, el enfoque RAG combina lo mejor de la recuperación de información y la generación de lenguaje natural para superar las limitaciones tradicionales de los LLMs. Su estructura flexible y escalable lo convierte en una herramienta poderosa para aplicaciones especializadas en dominios exigentes como la ciberseguridad, permitiendo generar respuestas precisas, actualizadas y contextualmente relevantes.

2.3. Sistemas SIEM

Uno de los grandes desafíos en el ámbito de la ciberseguridad es gestionar la abrumadora cantidad de datos que los sistemas generan cada día. Cada evento en un sistema digital, desde la conexión de un usuario hasta el acceso a un archivo, deja un registro. Esta acumulación masiva de datos tiene un gran potencial, pero también supone un problema

crítico: discernir rápidamente qué información es relevante y requiere atención inmediata. En muchas ocasiones, los equipos de seguridad enfrentan una sobrecarga de alertas que les impide actuar con eficacia, lo que aumenta el riesgo de pasar por alto amenazas significativas.

Los Sistemas de Información y Gestión de Eventos de seguridad, conocidos como SIEM, han surgido como una solución integral para este problema. Estos sistemas recopilan, analizan y correlacionan eventos de seguridad en tiempo real, proporcionando a los analistas una visión unificada del estado de los sistemas. Además de detectar amenazas, los SIEM automatizan la generación de alertas y, en algunos casos, ofrecen información clave para responder ante incidentes. Sin embargo, la gestión de tanta información sigue siendo un desafío.

En este contexto, surge la posibilidad de combinar los SIEM con modelos avanzados de RAG. A través de esta integración, los datos generados por un SIEM podrían enriquecerse con bases de conocimiento externas y catálogos de amenazas, proporcionando un análisis más profundo y recomendaciones específicas. Por ejemplo, en lugar de limitarse a señalar un evento sospechoso, un sistema SIEM-RAG podría contextualizar la amenaza dentro de un marco como MITRE ATT&CK, relacionarla con vulnerabilidades conocidas y proponer un curso de acción basado en regulaciones aplicables o estrategias de mitigación probadas. Este enfoque no solo mejoraría la relevancia de las alertas, sino que también reduciría significativamente la carga informativa, priorizando las amenazas más críticas y ofreciendo una respuesta más efectiva.

Para este proyecto, se propone un modelo en el que el sistema SIEM, apoyado por RAG, no solo recopile información del sistema monitoreado, sino que también integre datos externos, como bases de amenazas, guías normativas y configuraciones específicas del entorno. Este modelo tendría el potencial de analizar el estado de los sistemas en tiempo real, correlacionar eventos relevantes y, al mismo tiempo, proponer cursos de acción personalizados para mitigar amenazas. En este marco, los SIEM se convierten en algo más que sistemas de detección; evolucionan hacia herramientas de toma de decisiones automatizadas que mejoran la eficiencia y precisión del análisis de seguridad.

2.3.1. Wazuh

En el desarrollo de este estudio, se ha optado por utilizar Wazuh [1] como plataforma base para implementar esta propuesta. Aunque en el mercado existen otros SIEM ampliamente utilizados, como Splunk [38] o Elastic Security [39], la elección de Wazuh responde a su uso dentro del grupo de investigación al que pertenece este trabajo. Este contexto proporciona acceso a una infraestructura ya configurada, lo que facilita el desarrollo y la validación de las propuestas planteadas.

La arquitectura de Wazuh (Figura 2.3) combina un SIEM con un Sistema de Detección de Intrusiones (IDS), permitiendo recopilar y analizar datos de seguridad provenientes de múltiples fuentes, como registros de sistema, eventos y tráfico de red. Estos datos son después gestionados de manera centralizada por el Wazuh Manager, que trabaja en conjunto con agentes desplegados en los sistemas monitoreados. Estos agentes no solo recopilan información, sino que también pueden ejecutar acciones de respuesta, como actualizar reglas de firewall, fortaleciendo así la protección de los sistemas y su capacidad de respuesta ante incidentes.

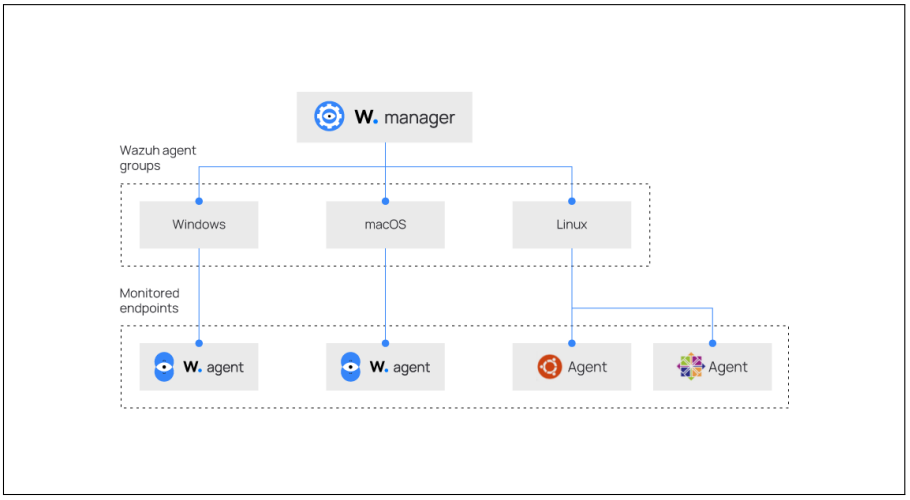

Figura 2.3: Arquitectura de Wazuh.
Imagen obtenida de [40].

Además de sus capacidades generales, Wazuh destaca por funciones avanzadas [41] como la detección de intrusiones mediante análisis de registros y eventos con reglas personalizables, y la gestión del cumplimiento normativo con informes alineados a estándares [42] como PCI DSS, HIPAA y GDPR. Incluye un módulo de respuesta activa para ejecutar acciones automáticas o manuales, como bloquear IPs, y ofrece herramientas de visualización integradas con Elasticsearch y Kibana. También permite análisis en tiempo real de logs y monitoreo de integridad de archivos, alertando sobre cambios sospechosos, y facilita la detección de vulnerabilidades mediante inventarios de software comparados con bases de datos CVE. Estas características convierten a Wazuh en una solución integral para la ciberseguridad.

En conclusión, la integración de un SIEM como Wazuh con un modelo RAG representa una evolución natural en el análisis de la seguridad informática. Al combinar la capacidad de recopilación de datos en tiempo real con un enfoque basado en RAG, se logra no solo mejorar la detección de amenazas, sino también optimizar la toma de decisiones y la priorización de acciones. Este proyecto busca demostrar cómo esta combinación puede transformar la manera en que se gestiona la seguridad en sistemas complejos, ofreciendo soluciones innovadoras y efectivas frente a los retos del entorno digital actual.

3. Estado del Arte

El análisis del estado del arte resulta fundamental para contextualizar la propuesta de este TFG dentro del marco de investigación existente. A través del estudio de trabajos previos se pueden identificar tanto las soluciones más avanzadas en la aplicación de modelos de lenguaje en ciberseguridad como las limitaciones aún no resueltas. Esta revisión permite comprender cómo las técnicas RAG, en combinación con LLMs, están siendo utilizadas actualmente para tareas como la detección de anomalías, la simulación de ataques o la automatización del análisis de eventos. A partir de este repaso crítico, se justifican las aportaciones particulares de este proyecto.

3.1. Trabajos relacionados

El uso combinado de LLMs y RAG (modelos LLM-RAG para futuras referencias) en ciberseguridad ha sido ampliamente estudiado en distintas aplicaciones, desde la simulación de ataques hasta la detección de anomalías y la optimización de la gestión de eventos de seguridad. Si bien estas investigaciones han sentado las bases para nuevas soluciones, este trabajo introduce un enfoque más integral que combina la detección y clasificación de amenazas con recuperación aumentada y la generación de respuestas accionables basadas en catálogos especializados. Uno de los estudios más relevantes en este ámbito es el realizado por Muhammad Mudassar et al. [43], que explora el uso de modelos de lenguaje para crear escenarios de entrenamiento dinámicos. Su enfoque aprovecha la capacidad generativa de los LLMs para simular ataques de manera realista, lo que resulta útil en entornos educativos. De manera complementaria, SECURE [44] desarrolla un benchmark para evaluar el desempeño de estos modelos en la extracción y análisis de información sobre amenazas de seguridad.

Más allá de la simulación y evaluación de modelos, otros estudios han abordado la mejora en la recuperación de información para consultas especializadas. MoRSE [45] presenta un chatbot basado en el modelo LLM-RAG diseñado para responder preguntas sobre ciberseguridad con mayor precisión. Su principal fortaleza radica en su capacidad para recuperar información estructurada y no estructurada de fuentes actualizadas, minimizando alucinaciones y mejorando la fiabilidad de sus respuestas. En la misma línea, RAGLog [46] emplea bases de datos vectoriales para almacenar logs normales y permitir que un LLM identifique anomalías mediante análisis semántico. Este enfoque guarda una estrecha relación con la solución planteada en este trabajo, ya que ambos buscan mejorar la detección de amenazas en registros de seguridad mediante el modelo LLM-RAG. No obstante, mientras RAGLog se centra exclusivamente en la identificación de anomalías, el sistema propuesto en este TFG pretende ampliar el concepto, incorporando una priorización de amenazas y la generación de recomendaciones de acción, proporcionando una solución más completa para la ciberdefensa.

Si se consideran únicamente los modelos de lenguaje sin recuperación aumentada, existen diversas investigaciones que analizan su aplicación en ciberseguridad. Jiarui Li et al. [47] estudian cómo estos modelos pueden mejorar la eficiencia operativa en los Centros de Operaciones de Seguridad (SOC), centrándose en la clasificación automática de alertas para reducir la carga de trabajo de los analistas y optimizar la toma de decisiones en entornos SIEM. De manera similar, los estudios de Mikko Lempinen et al. [6][48] y Cyber Sentinel [49] exploran la incorporación de asistentes conversacionales en la gestión de eventos de seguridad, utilizando modelos de lenguaje para interpretar registros, responder consultas y ejecutar acciones de mitigación. En esta línea, Jie Zhang et al. [50] presentan un análisis exhaustivo sobre la aplicación de los LLMs en ciberseguridad, abarcando más de 300 estudios previos. Su trabajo destaca el uso de estos modelos en tareas como la detección de vulnerabilidades, la generación de inteligencia de amenazas y el análisis de anomalías en registros de sistemas. Además, identifican desafíos clave como la escalabilidad, la susceptibilidad a ataques adversariales y la necesidad de mejorar la evaluación de su desempeño en entornos reales.

3.2. Aportaciones de este trabajo

Con todo ello, este TFG se diferencia de los anteriores al centrarse en la integración de RAG con catálogos especializados para la generación de respuestas más precisas y aplicables en el contexto de la ciberdefensa. Al integrar un LLM con información estructurada sobre amenazas y cursos de acción, el sistema no solo identifica eventos de seguridad relevantes, sino que también proporciona recomendaciones fundamentadas para su mitigación. Esta capacidad de contextualización mejora la fiabilidad de las respuestas y reduce la dependencia de la intervención humana en la interpretación de alertas.

El análisis de estas investigaciones permite contextualizar este TFG dentro del estado del arte en la integración de modelos LLM-RAG en ciberseguridad. Su principal aporte radica en la combinación de varios elementos innovadores que no han sido abordados de manera conjunta en la literatura. Por un lado, el uso de un generador de logs simulados proporciona un entorno controlado para evaluar el rendimiento del sistema. Por otro, la integración con un SIEM garantiza que la solución sea aplicable en escenarios reales. Finalmente, la diferencia clave en esta propuesta es que no se limita a la detección de anomalías, sino que introduce un mecanismo de priorización de amenazas y generación de COAs, diferenciándose así de los estudios previos.

Otro aspecto clave que distingue nuestra solución de las expuestas en los artículos referidos anteriormente, es la elección de Llama 3.1 como modelo base. En contraste con la mayoría de investigaciones encontradas, que emplean modelos propietarios como GPT-4, al optar por un modelo de código abierto no solo obtenemos mayor flexibilidad en la implementación, sino que también facilitamos su adaptación a diferentes entornos sin depender de soluciones comerciales. Esta decisión refuerza la accesibilidad y replicabilidad del sistema, favoreciendo su adopción en contextos donde el uso de modelos cerrados podría estar limitado por restricciones de privacidad o costos.

En definitiva, este proyecto se basa en avances previos, pero introduce una integración más completa al incorporar recuperación aumentada con catálogos especializados, priorización de amenazas y generación de COAs. Esta combinación permite desarrollar un sistema más fiable y eficiente para la automatización de la ciberdefensa en entornos SIEM.

4. Desarrollo

En este apartado, se describe la arquitectura general del sistema, las herramientas utilizadas y los detalles clave de la implementación. El objetivo principal es desarrollar un sistema que integre IA generativa con un enfoque de Retrieval Augmented Generation para analizar registros generados por un SIEM, específicamente Wazuh.

4.1. Arquitectura

La arquitectura del sistema desarrollado en este TFG se fundamenta en la integración de modelos de inteligencia artificial generativa dentro de un entorno de ciberseguridad, mediante un enfoque RAG. Su objetivo principal es mejorar la detección, priorización y respuesta ante amenazas identificadas en los logs generados por un sistema SIEM.

Este sistema no pretende sustituir a un SIEM tradicional, sino complementarlo mediante una capa adicional de inteligencia generativa basada en el enfoque RAG. Mientras el SIEM se encarga de recopilar, correlacionar y alertar sobre eventos de seguridad, el sistema propuesto actúa a continuación, analizando dichas alertas mediante un modelo LLM enriquecido con conocimiento contextual externo. Esta integración permite clasificar las amenazas según su nivel de criticidad y proponer cursos de acción personalizados, transformando así al SIEM en una herramienta no solo de detección, sino también de recomendación inteligente y asistida, que mejora significativamente la capacidad de respuesta ante incidentes de seguridad.

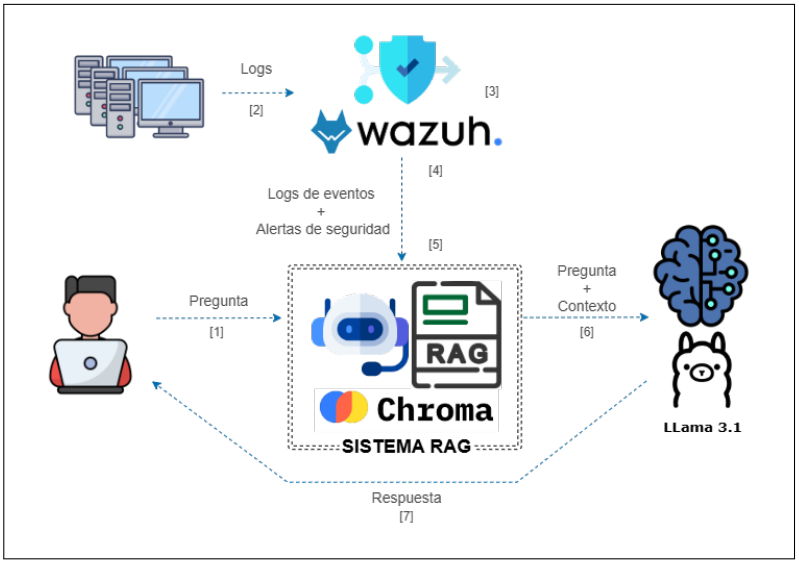

Figura 4.1: Diagrama de arquitectura del sistema

Como se muestra en la Figura 4.1, la arquitectura propuesta combina tres componentes principales: el sistema SIEM (en nuestro caso, Wazuh), que recopila y analiza eventos de seguridad; un motor de recuperación de información (Chroma DB), que consulta una base de conocimiento contextual; y un modelo de lenguaje (LLaMA 3.1), encargado de generar recomendaciones accionables en función del análisis. Este diseño permite automatizar la interpretación de alertas y proponer respuestas adaptadas, optimizando así los tiempos de reacción frente a ciberamenazas.

4.1.1. Flujo de operación del sistema global

A continuación, se detalla el flujo de trabajo del sistema descrito en la Figura 4.1, el cual describe paso a paso cómo se gestionan los eventos de seguridad desde su detección inicial hasta la generación de una respuesta accionable.

1. **Recepción de la consulta del usuario**

   El flujo de operación del sistema comienza con la formulación de una consulta por parte del usuario. Esta consulta, que puede referirse a la existencia de amenazas en los logs o a recomendaciones específicas de mitigación, no se somete a un proceso de vectorización como ocurre con otros elementos del sistema. En su lugar, se introduce directamente en el prompt dirigido al modelo LLM, permitiendo que el modelo tenga una comprensión explícita de la necesidad informativa del usuario. Cabe destacar que esta funcionalidad no constituye el foco principal del presente TFG, por lo que la pregunta ha sido incorporada manualmente dentro del pipeline como un elemento estático de prueba, con el objetivo de facilitar la evaluación del sistema en escenarios concretos.

2. **Captura de logs**

   De forma paralela, se generan los logs que provienen de distintos dispositivos y sistemas dentro de una infraestructura de TI. Estos registros contienen información clave sobre eventos de seguridad, como intentos de acceso, modificaciones en archivos críticos, tráfico inusual en la red o cualquier otro comportamiento que pueda ser indicativo de una posible amenaza. Estos eventos son recopilados y gestionados por Wazuh, que actúa como el núcleo de monitorización del sistema.

3. **Análisis inicial en un SIEM**

   Wazuh procesa los registros de eventos mediante reglas previamente definidas y correlaciones avanzadas para detectar patrones sospechosos. En caso de identificar una posible amenaza, el sistema genera una alerta que contiene detalles específicos sobre el evento detectado, su nivel de riesgo y otros metadatos relevantes. No obstante, la detección por sí sola no es suficiente para una respuesta efectiva, especialmente en entornos donde la cantidad de alertas generadas puede ser abrumadora.

4. **Envío de la alerta al sistema RAG**

   Aquí es donde entra en juego el sistema RAG, que se encarga de contextualizar las alertas generadas por Wazuh con información adicional proveniente de fuentes externas. Para ello, los logs se transforman en representaciones vectoriales (embeddings) mediante el LLM, lo que permite realizar una búsqueda por similitud semántica en la base de conocimientos de ChromaDB [33] previamente indexada. Esta estrategia

facilita la recuperación de fragmentos relevantes que contienen información contextualizada sobre amenazas, técnicas de ataque o medidas de mitigación. A diferencia de un SIEM tradicional, que se limita a correlacionar eventos dentro de su propio conjunto de datos, la implementación de un sistema RAG permite enriquecer el análisis con fuentes externas como el framework MITRE ATT&CK [2], bases de datos de vulnerabilidades conocidas (CVE [51]), informes de inteligencia de amenazas y casos previos documentados en la literatura de ciberseguridad.

5. **Análisis con LLM**

   Para llevar a cabo este análisis avanzado, el sistema RAG toma como entrada los logs y alertas generados por Wazuh y formula una consulta optimizada para el modelo de lenguaje Llama 3.1. Esta consulta no solo incluye la información básica del evento detectado, sino que además incorpora la pregunta del usuario y el contexto adicional recuperado de fuentes confiables. De esta forma, el modelo de lenguaje no se limita a interpretar los datos de manera aislada, sino que los analiza dentro de un marco de referencia mucho más amplio, lo que permite reducir falsos positivos y mejorar la precisión del análisis.

6. **Generación de COAs**

   Una vez que Llama 3.1 recibe la consulta enriquecida, el modelo genera una respuesta detallada en la que no solo se clasifica la amenaza según su nivel de criticidad, sino que también se proponen Cursos de Acción (COAs) para mitigarla. Este proceso se basa en la definición de un prompt cuidadosamente estructurado, que combina los logs analizados con el contexto recuperado, e indica al modelo el formato y tipo de respuesta esperada. Gracias a este diseño, se optimiza la generación de recomendaciones claras, precisas y accionables. Estas pueden incluir desde la actualización de reglas de firewall y la aplicación de parches de seguridad hasta cambios en políticas de acceso o segmentación de red, dependiendo del tipo de amenaza identificada.

7. **Entrega de la respuesta al usuario**

   Finalmente, la información generada por el modelo se devuelve al usuario en un formato claro y estructurado, facilitando la toma de decisiones. De esta manera, el sistema no solo permite detectar amenazas con mayor precisión, sino que también proporciona un mecanismo automatizado para priorizar y responder de manera efectiva a los incidentes de seguridad.

Durante el resto del capítulo se desarrollará la implementación del sistema RAG creado.

4.2. Herramientas Utilizadas

El desarrollo del sistema se apoya en varias herramientas de software y *frameworks*, entre ellas:

Herramienta	Propósito en el sistema
LLaMA 3.1 [52]	Modelo de lenguaje para el análisis y clasificación de amenazas, así como la generación de recomendaciones.
Python [53]	Lenguaje de programación principal para la integración de módulos, procesamiento de datos y generación de respuestas.
LangChain [54]	Framework para la implementación del flujo de RAG, facilitando la integración del modelo LLM con bases de conocimiento externas.
ChromaDB [33]	Base de datos vectorial utilizada para la indexación y recuperación eficiente de documentos relevantes en el sistema RAG.
LangSmith [55]	Herramienta para la evaluación, depuración y optimización del flujo de interacción entre el LLM y el sistema.
Docker [56]	Contenerización de la LLM para garantizar su portabilidad y reproducibilidad en distintos entornos.
Ollama [57]	Plataforma para la ejecución local de modelos de lenguaje, optimizando la inferencia de LLaMA 3.1 sin depender de la nube.

Tabla 4.1: Resumen de herramientas utilizadas y su propósito en el sistema.

4.2.1. Selección de Tecnologías para el Sistema RAG

Para implementar este sistema, se ha optado por utilizar el modelo pre-entrenado LLaMA 3.1:8B [52]. Este modelo incluye una versión para la generación de embeddings y otra para la generación de texto. Su funcionalidad de embeddings, con 8 billones de parámetros, ofrece una base sólida para representar semánticamente los fragmentos y consultas, garantizando una alta precisión y adaptabilidad.

Training Data	Input modalities	Output modalities	Context length	GQA	Token count	Knowledge cutoff
Llama 3.1 8B (text only)	Multilingual Text	Multilingual Text and code	128k	Yes	15T+	December 2023

Tabla 4.2: Características del modelo Llama 3.1:8B

Entre las razones que justifican su elección se encuentran:

- Escalabilidad: Gracias a su arquitectura avanzada, LLaMA 3.1 supera a modelos más pequeños en precisión y capacidad de representación.
- Adaptabilidad Contextual: Los embeddings generados son dinámicos, ajustándose al contexto específico del sistema.
- Capacidad Multilingüe: Este modelo permite trabajar con fuentes de información en múltiples idiomas, un aspecto clave para sistemas que integran datos diversos.

En cuanto al mecanismo de recuperación, se ha seleccionado el algoritmo *k-Nearest Neighbors* (k-NN), lo que permite localizar los k fragmentos más similares en el espacio vectorial. La elección de este algoritmo se fundamenta en su simplicidad, eficacia y adaptabilidad a

espacios multidimensionales, tal y como señalan Li et al. [58]. Al tratarse de un método no paramétrico, k-NN no requiere una fase previa de entrenamiento, lo que lo convierte en una solución robusta para sistemas dinámicos como el propuesto, donde los vectores a consultar pueden variar frecuentemente. Además, su capacidad para trabajar con diferentes métricas de distancia lo hace especialmente adecuado para tareas de recuperación semántica en entornos RAG.

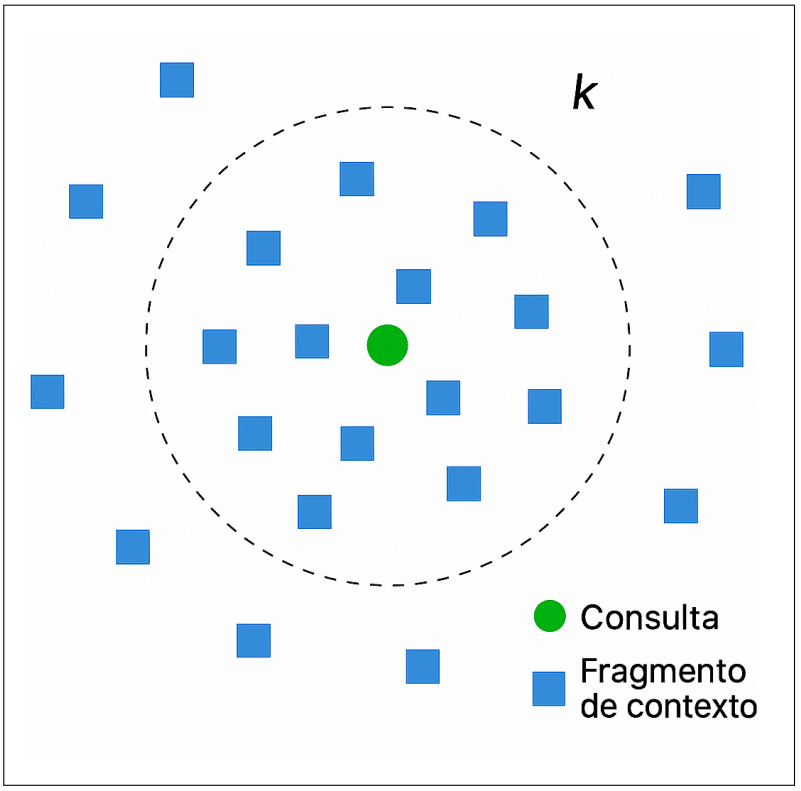

Figura 4.2: Algoritmo k-NN

Ambas tecnologías, LLaMA 3.1 y k-NN, han sido seleccionadas no solo por su rendimiento, sino también por su amplia aceptación en la comunidad de investigación, con numerosos estudios que respaldan su uso en sistemas RAG como [59], [60] y [61].

4.3. Implementación del Sistema RAG

La implementación del sistema RAG en este proyecto busca mejorar la capacidad de detección y respuesta ante amenazas en el entorno de ciberseguridad. Para ello, se ha diseñado un flujo de trabajo en el que los eventos de seguridad generados por Wazuh se combinan con información relevante extraída de una base de conocimientos especializada, permitiendo que el modelo de lenguaje Llama 3.1 analice los incidentes con mayor precisión. El sistema RAG no solo recupera información útil para enriquecer el contexto de los logs analizados, sino que también genera respuestas estructuradas que incluyen un diagnóstico del evento detectado y recomendaciones para su mitigación. En este apartado, se detallan los principales componentes del sistema RAG, excluyendo la fase de vectorización y almacenamiento de embeddings, que ya ha sido descrita en secciones anteriores.

4.3.1. Base de Conocimientos

Para que el sistema RAG pueda proporcionar análisis precisos y contextualizados, es fundamental contar con una base de conocimientos sólida y especializada. En este proyecto, dicha base se construye a partir de documentos técnicos, catálogos de amenazas y marcos normativos ampliamente reconocidos en el ámbito de la ciberseguridad.

Toda esta información se ha procesado mediante técnicas de segmentación semántica y generación de *embeddings* utilizando el modelo LLaMA 3.1:8B, lo que permite su integración eficiente en el sistema y su posterior recuperación contextual. Es importante destacar que esta base de conocimientos se genera de forma previa e independiente al *pipeline* de recuperación y generación, permitiendo su reutilización, actualización y mantenimiento sin necesidad de reentrenar o modificar el flujo principal del sistema RAG.

Fuentes de Información Utilizadas

Para construir una base de conocimientos sólida y pertinente, se ha seguido una metodología basada en la selección de documentos reconocidos internacionalmente por su relevancia en la identificación, análisis y mitigación de amenazas de ciberseguridad. Se han priorizado estándares, marcos y catálogos elaborados por entidades de referencia como el NIST y MITRE, ya que proporcionan tanto una visión estructurada de las TTPs empleadas por actores maliciosos, como directrices defensivas aplicables a entornos reales. La selección se ha orientado hacia fuentes que no solo describen comportamientos ofensivos, sino que también ofrecen medidas concretas para su detección y contención, asegurando así que el sistema RAG pueda generar respuestas contextualizadas, fundamentadas y alineadas con las mejores prácticas del sector.

Aunque existen otras fuentes igualmente valiosas en el ámbito de la ciberseguridad, como las guías técnicas de la Agencia de la Unión Europea para la Ciberseguridad (ENISA) [62], la documentación de CIS (Center for Internet Security) [63], los reportes de amenazas de empresas privadas (como Mandiant [64], Palo Alto [65] o CrowdStrike [66]) o las bases de datos de vulnerabilidades como NVD (National Vulnerability Database) [67], estas no han sido incluidas en la presente versión del sistema por diversos motivos. En primer lugar, muchas de estas fuentes presentan un elevado volumen de datos o una estructura técnica que requeriría procesos adicionales de limpieza, segmentación o estandarización. En segundo lugar, la capacidad computacional disponible ha impuesto ciertas limitaciones en cuanto al tamaño total de la base vectorial y el número de documentos procesables simultáneamente, lo que ha motivado la priorización de fuentes más estructuradas y directamente aplicables al contexto RAG. Finalmente, se ha buscado mantener un equilibrio entre diversidad, aplicabilidad y mantenimiento, seleccionando fuentes que puedan ser actualizadas de forma sencilla y que proporcionen cobertura suficiente tanto en la identificación de amenazas como en la respuesta defensiva. En futuras iteraciones del sistema, la incorporación progresiva de estas fuentes adicionales podría aumentar su precisión y cobertura.

- **Marco NIST (National Institute of Standards and Technology) [68]**

  El NIST es una entidad de referencia a nivel global en la elaboración de estándares en seguridad informática, la cual proporciona directrices y buenas prácticas para gestionar y reducir riesgos relacionados con la seguridad de la información. Se han incorporado tres de sus publicaciones especialmente relevantes:

- **NIST SP 800-53 [69] – Security and Privacy Controls for Information Systems and Organizations:** establece un catálogo exhaustivo de controles de seguridad clasificados en familias (como acceso, auditoría, respuesta a incidentes, etc.), diseñados para proteger la confidencialidad, integridad y disponibilidad de los sistemas de información.
- **NIST SP 800-61 [70] – Computer Security Incident Handling Guide:** proporciona una guía completa para el manejo de incidentes de seguridad, desde la preparación inicial hasta la recuperación y análisis post-incidente. Es esencial para comprender los tipos de incidentes que pueden aparecer en los logs del sistema.
- **NIST SP 800-92 [71] – Guide to Computer Security Log Management:** profundiza en la importancia de los logs en la detección de amenazas y cumplimiento normativo, proponiendo buenas prácticas para su recolección, almacenamiento y análisis.

Para realizar los embeddings, estos tres documentos han sido procesados como PDF, convertidos a texto, fragmentados en unidades semánticas (fragmentos de 1000 caracteres con solapamiento) y vectorizados con el modelo de embedding de LLaMA 3.1.

- **MITRE ATT&CK [2]**

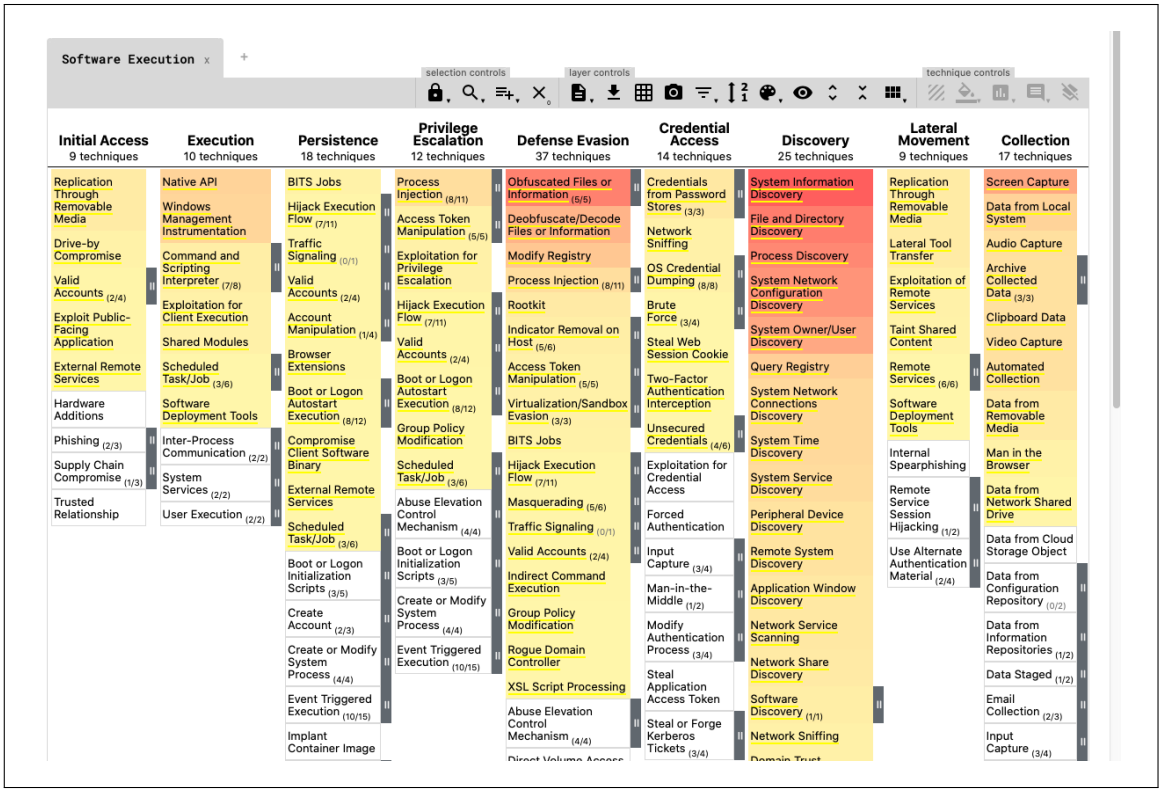

Figura 4.3: Matriz Mitre Attack

Junto a los documentos del NIST, el sistema incorpora información procedente de distintos marcos desarrollados por MITRE. En primer lugar, se ha utilizado el framework de a matriz ATT&CK (Figura 4.3), concretamente su versión *enterprise* en

formato JSON. Este catálogo describe tácticas, técnicas y procedimientos (TTPs) empleados por actores maliciosos en campañas reales de ataque, permitiendo establecer correlaciones entre los eventos detectados y patrones de comportamiento conocidos. Cada técnica ha sido extraída, formateada en texto explicativo y posteriormente vectorizada como parte del cuerpo de recuperación del sistema.

- **MITRE D3FEND [72]**

  Complementando la perspectiva ofensiva del ATT&CK, el proyecto también incorpora el marco MITRE D3FEND, que reúne técnicas defensivas diseñadas para mitigar las amenazas descritas en ATT&CK. Este catálogo, originalmente en formato hoja de cálculo y convertido a JSON, contiene información jerárquica sobre tácticas y técnicas defensivas, junto a definiciones claras y estructuradas. Cada entrada ha sido tratada como una unidad documental separada, lista para su recuperación semántica.

- **MITRE CAPEC [73] y CWE [74]**

  El CAPEC (Common Attack Pattern Enumeration and Classification), proporciona descripciones detalladas de patrones comunes de ataque, incluyendo sus mecanismos, objetivos y posibles mitigaciones. Es especialmente útil para contextualizar técnicas ofensivas desde una perspectiva de ingeniería del atacante.

  Por otro lado, el CWE (Common Weakness Enumeration) cataloga debilidades en software que pueden ser explotadas por actores maliciosos. Incluye información sobre la naturaleza de la vulnerabilidad, su probabilidad de explotación, el estado en el que se encuentra y los patrones de ataque relacionados.

  Tanto CAPEC como CWE han sido procesados desde archivos JSON oficiales, estructurando cada entrada como documento textual enriquecido con los campos más relevantes antes de ser embebido con el modelo LLaMA.

Cada una de estas fuentes ha sido cuidadosamente preprocesada y vectorizada para garantizar su compatibilidad con el módulo de recuperación semántica del sistema RAG. Gracias a esta base de conocimiento estructurada y especializada, el sistema es capaz de ofrecer respuestas contextualizadas, precisas y alineadas con las mejores prácticas en ciberseguridad.

4.3.2. Vectorización y Almacenamiento

En el proceso de implementación del sistema RAG, una de las fases fundamentales es la vectorización y almacenamiento de los datos (embeddings). Esta etapa permite transformar la información en representaciones matemáticas (vectores) que facilitan su análisis y posterior recuperación, mediante una búsqueda por semejanza de vectores semánticos. Para comprender mejor el comportamiento de los embeddings y su impacto en el sistema, se han implementado técnicas de visualización y *clustering* que permiten explorar su distribución y estructura.

Generación, Almacenamiento y Visualización de Embeddings

Para analizar cómo los datos son representados en el espacio de embeddings, se han extraído vectores a partir de las diferentes fuentes de información utilizadas en este trabajo:

- **Contexto de Ciberseguridad:** Información extraída de documentación técnica, bases de datos de amenazas y conocimiento experto.

- **Logs de Eventos de Seguridad:** Registros generados en el SIEM en cada uno de los casos de prueba (CALDERA y CTF).

Cada uno de estos elementos es transformado en embeddings mediante el modelo de lenguaje LLaMA 3.1, lo que permite visualizar cómo se estructuran las relaciones entre los distintos eventos y conceptos de ciberseguridad en un espacio vectorial.

El flujo de trabajo seguido para la generación, almacenamiento y visualización de los embeddings en el sistema es el siguiente:

1. **Extracción de Embeddings:** Los eventos registrados en el SIEM son procesados y transformados en vectores mediante el modelo de embedding LLaMA 3.1.

2. **Almacenamiento en ChromaDB:** Los embeddings se almacenan en ChromaDB [33], una base de datos vectorial que facilita su acceso y procesamiento posterior.

3. **Recuperación de Datos:** Los embeddings almacenados pueden ser consultados para ser utilizados en análisis y clustering.

4. **Visualización de Embeddings:** Se emplean técnicas de reducción de dimensionalidad para analizar la distribución de los eventos y representar gráficamente su estructura. Esto nos permite comprobar las dimensiones semánticas de los *embeddings*.

Visualización de Embeddings

Dado que los embeddings se encuentran en un espacio de alta dimensión, se han aplicado técnicas de reducción de dimensionalidad para representar los datos en un espacio 2D. Esto facilita la interpretación de las relaciones entre los embeddings y permite identificar patrones en la información.

Para ello, se han utilizado dos métodos principales:

- **PCA (Principal Component Analysis):** Técnica, visible en la Figura 4.4, que permite proyectar los datos en un espacio de menor dimensión preservando la mayor cantidad posible de varianza

- **UMAP (Uniform Manifold Approximation and Projection):** Técnica avanzada, representada en la Figura 4.5, que mantiene la estructura de los datos y es especialmente útil para análisis de *clustering*.

Con estas gráficas podemos identificar patrones y diferencias en la distribución de los embeddings según su origen, mostrando cómo se estructuran los conceptos clave del contexto en el espacio vectorial y cómo se agrupan los eventos de seguridad generados en la simulación a partir de los logs y registros del CTF.

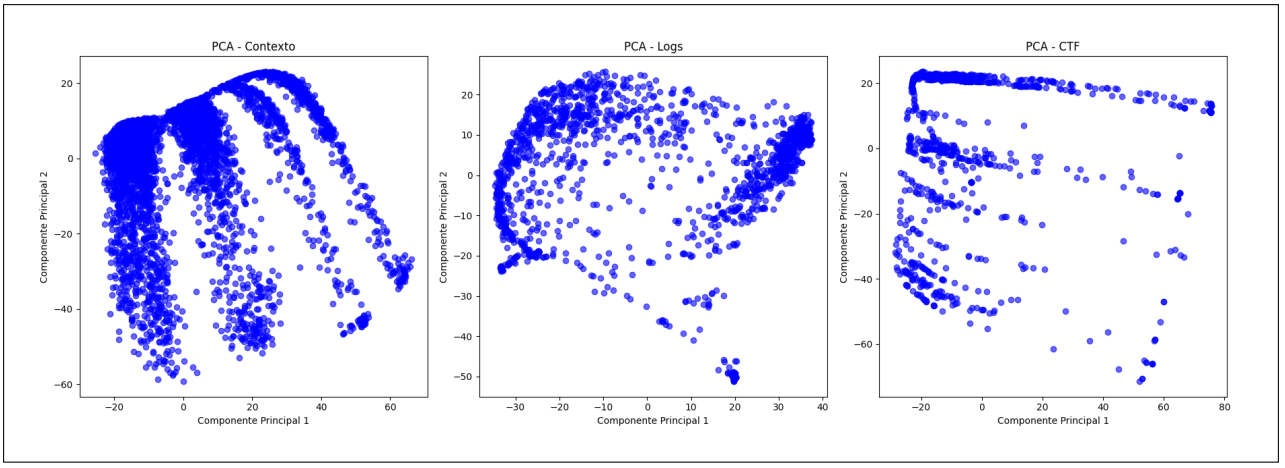

Figura 4.4: Representación de los embeddings con PCA

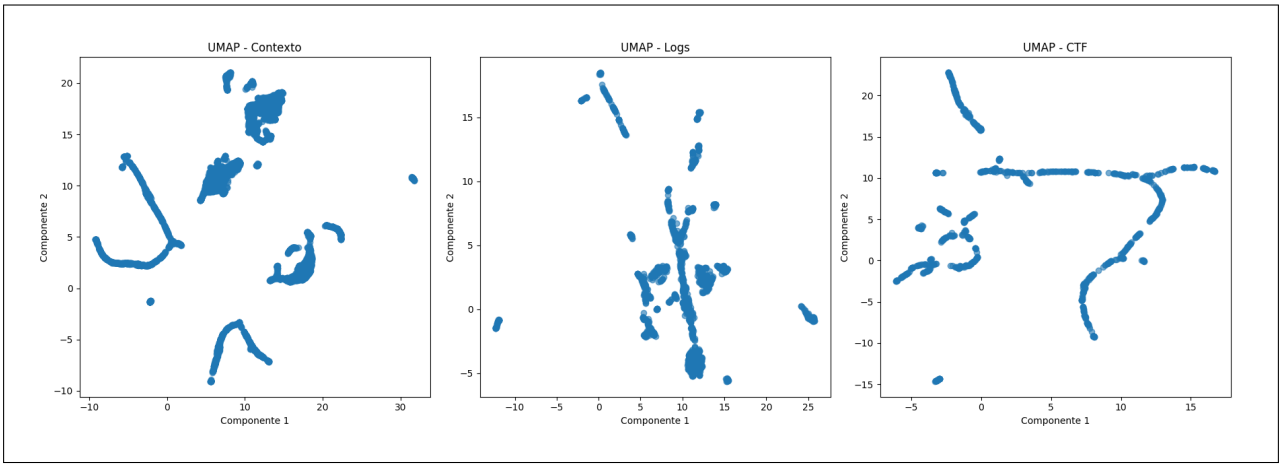

Figura 4.5: Representación de los embeddings con UMAP

Interpretación de Embeddings mediante Clustering

El clustering es una técnica de aprendizaje no supervisado que permite agrupar elementos similares dentro de un conjunto de datos sin necesidad de etiquetas predefinidas. En este caso, se ha aplicado esta técnica sobre los embeddings del contexto para identificar patrones y estructuras dentro de la información utilizada por el sistema RAG.

Gracias al clustering, es posible descubrir relaciones entre conceptos, estructurar mejor la información y optimizar la recuperación de datos relevantes. Para evaluar la segmentación de los embeddings del contexto, se han seleccionado cuatro algoritmos de clustering represenatdos en la Figura 4.6:

- **KMeans:** Método eficiente cuando se conoce el número de grupos esperados.
- **HDBSCAN:** Detecta *clusters* de distintos tamaños sin necesidad de definir k.
- **GMM (Gaussian Mixture Model):** Permite *clusters* con formas elípticas, útil en contextos con superposición de información.
- **OPTICS:** Similar a DBSCAN, pero con mejor detección de *clusters* de diferentes densidades.

Para comparar la calidad de los agrupamientos obtenidos, se han utilizado dos métricas estándar en entornos de clustering:

- **Silhouette Score:** Indica la coherencia interna de los clusters (valores más altos son mejores).
- **Davies-Bouldin Score:** Evalúa la separación entre clusters (valores más bajos indican mejor segmentación).

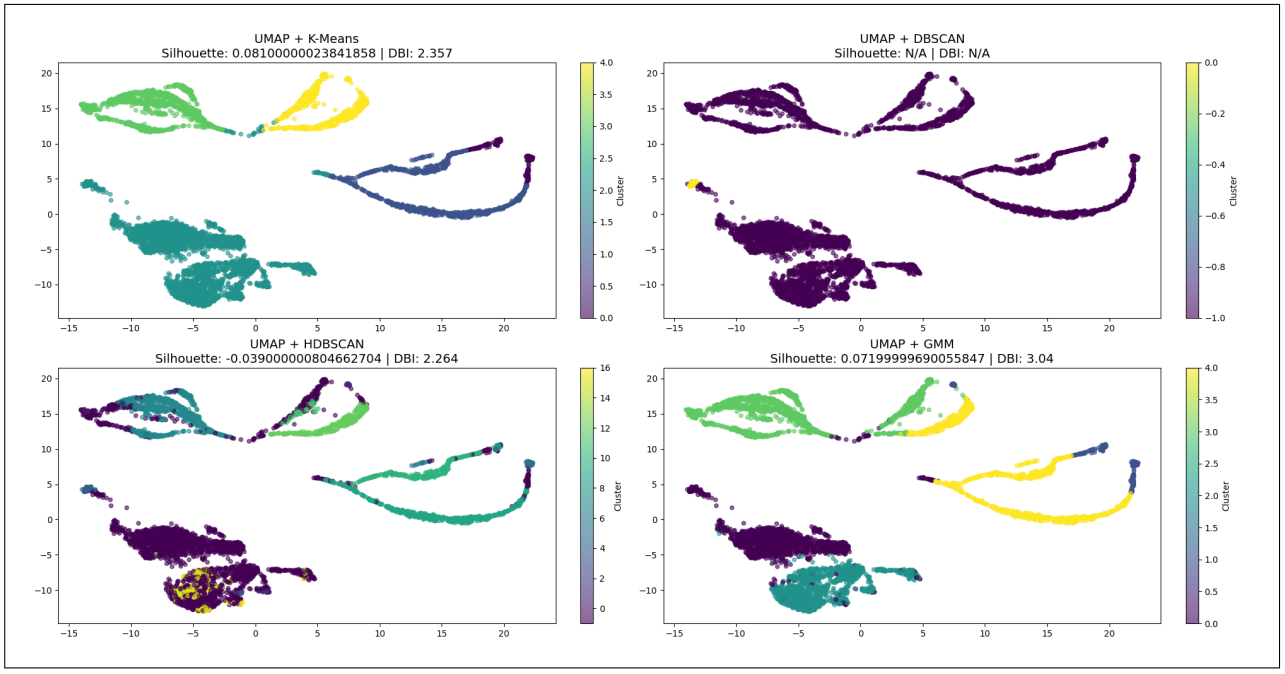

Figura 4.6: Distribución de los embeddings del contexto

Los resultados obtenidos (Tabla 4.3) permiten extraer varias conclusiones sobre la segmentación de los embeddings del contexto:

- **KMeans** y **GMM** generan agrupaciones bien definidas, lo que los hace adecuados cuando se busca una estructura clara y predefinida.
- **HDBSCAN** es útil para identificar relaciones emergentes sin necesidad de establecer un número fijo de clusters, proporcionando mayor flexibilidad.
- **DBSCAN** detecta clusters con variaciones en densidad, aunque requiere ajustes adicionales para optimizar su desempeño.

Método	Silhouette Score	Davies-Bouldin Score
KMeans	0.263	1.539
HDBSCAN	0.273	1.928
GMM	0.263	1.539
DBSCAN	0.189	0.65

Tabla 4.3: Evaluación de los algoritmos de clustering aplicados a los embeddings del contexto

En términos generales, este análisis contribuye a mejorar la organización y acceso a la información dentro del sistema RAG, facilitando la contextualización de eventos de seguridad y proporcionando recomendaciones más precisas en la detección de amenazas.

4.3.3. Consulta y Generación de Respuestas

El sistema propuesto en este TFG no solo recupera información de la base de conocimientos, sino que también la integra en la respuesta final generada por el modelo de lenguaje LLM. Este proceso se basa en una secuencia de pasos que permiten contextualizar los logs detectados en el SIEM y proporcionar recomendaciones de mitigación adecuadas.

Flujo del Proceso de Consulta y Respuesta

En resumen, el flujo de la generación de respuesta es el siguiente:

1. **Recuperación de logs:** Los registros generados en el SIEM son extraídos y preparados para su análisis.
2. **Búsqueda de información en la base de conocimientos:** Con estos logs, se consulta la base de embeddings almacenada en ChromaDB para recuperar fragmentos de información relevantes en función de su similitud semántica.
3. **Construcción de la consulta para LLaMA 3.1:** La información relevante se pasa como contexto y se genera un prompt estructurado que combina los logs detectados con esta información recuperada de la base de conocimientos. Este prompt incluye el contexto necesario para que el modelo pueda interpretar correctamente los eventos y ofrecer una respuesta fundamentada.
4. **Análisis y generación de respuesta con LLaMA 3.1:** El modelo LLM procesa la consulta y genera una respuesta estructurada, detallando:
   - Si hay una amenaza detectada.
   - El nivel de riesgo asociado.
   - Una explicación técnica del evento.
   - COAs recomendados para mitigar la amenaza.
5. **Entrega de la respuesta al usuario:** La respuesta final se presenta al usuario de manera clara y estructurada para facilitar la toma de decisiones por parte del analista de seguridad.

Búsqueda Semántica de Información

El sistema implementa una estrategia de búsqueda semántica basada en la transformación de los logs detectados en el SIEM en representaciones vectoriales mediante embeddings generados con el modelo LLaMA 3.1. Estos vectores son comparados con los existentes en la base de datos vectorial ChromaDB para identificar los fragmentos de conocimiento más cercanos semánticamente. El funcionamiento de esta búsqueda semántica se encuentra representado en la Figura 4.7.

Tal y como se ha mencionado previamente, para llevar a cabo esta búsqueda se ha seleccionado el algoritmo k-Nearest Neighbors (k-NN). En este caso se ha elegido un valor de $k = 10$ por estar dentro del rango óptimo identificado en la literatura. En particular, el estudio comparativo de métodos RAG presentado en [75] muestra que valores de k entre

5 y 10 proporcionan una recuperación efectiva y precisa, mientras que valores superiores a 15 tienden a introducir fragmentos redundantes o irrelevantes, afectando negativamente a la fidelidad y utilidad de las respuestas generadas. De este modo, el valor seleccionado garantiza un equilibrio adecuado entre cobertura semántica y eficiencia contextual.

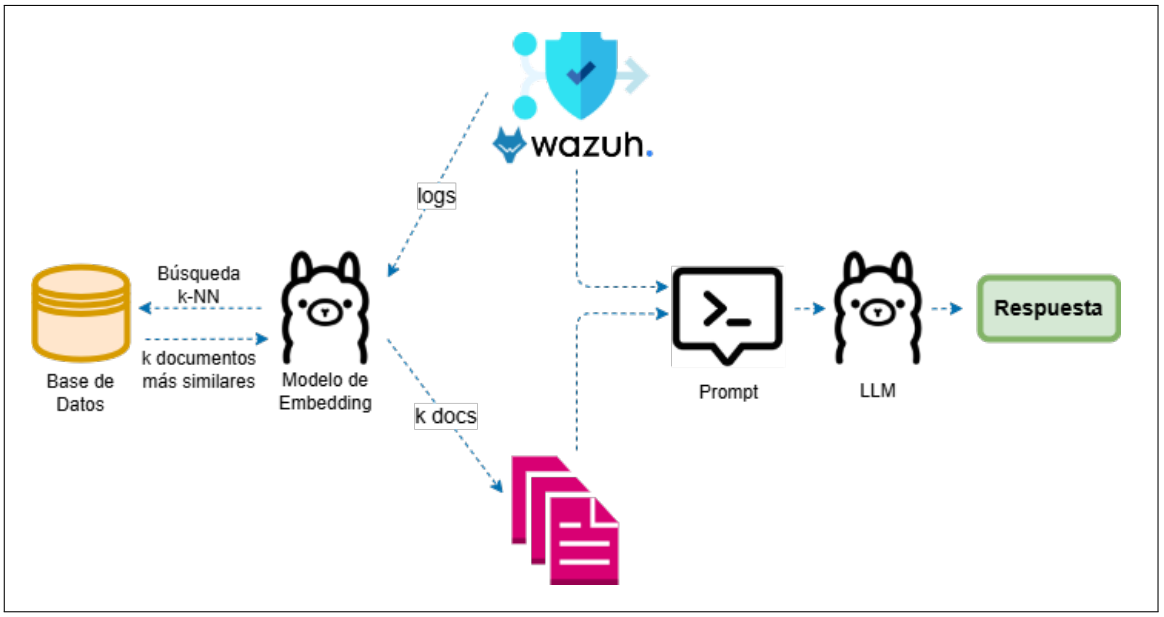

Figura 4.7: Búsqueda semántica del sistema

Respecto a la métrica utilizada, en este caso se ha seleccionado la *similitud del coseno* (*cosine similarity*), ampliamente empleada en el ámbito del procesamiento de lenguaje natural [75] por su capacidad para evaluar la orientación entre vectores, sin verse afectada por su magnitud. Esta propiedad la hace particularmente útil en embeddings textuales, donde lo relevante es la dirección semántica del vector y no su escala. Además, esta métrica ha demostrado ofrecer un buen equilibrio entre eficiencia computacional y calidad en la recuperación de contexto en arquitecturas RAG estándar.

Los fragmentos resultantes de esta búsqueda son utilizados posteriormente como contexto en la construcción del prompt para el modelo de lenguaje. De este modo, se proporciona al LLM información adicional específica y relevante, que mejora la precisión y fundamentación de las respuestas generadas sin necesidad de reentrenar el modelo.

Construcción del Prompt para el Modelo LLM

Para garantizar que LLaMA 3.1 genere respuestas precisas, útiles y estructuradas, se ha diseñado cuidadosamente un prompt optimizado, mostrado en la Figura 4.8. Este prompt constituye la interfaz entre los datos obtenidos por el sistema y el modelo de lenguaje, definiendo de forma explícita la tarea que debe realizar, el formato de la respuesta y la forma de utilizar el contexto recuperado.

El diseño del prompt sigue las mejores prácticas en ingeniería de instrucciones para modelos LLM [**?**], en particular en entornos críticos como la ciberseguridad, donde se requiere que las respuestas sean verificables, claras y alineadas con la información disponible. El prompt incluye secciones diferenciadas para los logs del SIEM, el contexto relevante recuperado mediante búsqueda semántica, y la consulta formulada por el usuario. Asimismo,

se especifican reglas que instruyen al modelo a dar prioridad a los datos empíricos de los logs y utilizar el contexto únicamente como refuerzo, evitando así alucinaciones o inferencias infundadas.

```
def setup_model_and_prompt():
    """Configura el modelo LLM y el prompt."""
    template = """
    Eres un asistente experto en ciberseguridad, especializado en la detección de amenazas y
    respuesta a incidentes.

    Se te proporcionan los siguientes datos:
    1. LOGS: Información extraída del sistema SIEM.
    2. CONTEXTO: Información relevante obtenida de bases de datos y documentación de seguridad.
    3. PREGUNTA: Un usuario solicita un análisis de seguridad y recomendaciones de mitigación.

    Tu tarea es:
    - Analiza los logs de forma autónoma, identificando cualquier posible actividad sospechosa sin
    depender del contexto.
    - A continuación, utiliza el contexto proporcionado únicamente para reforzar y justificar tus
    hallazgos, vinculándolos a evidencias documentadas.
    - Responde utilizando exactamente la estructura que se indica a continuación. No incluyas
    etiquetas entre corchetes ([]), ya que son sólo aclaraciones para ti.

    Datos proporcionados:
    <LOGS>
        {input_logs}
    </LOGS>
    <CONTEXTO>
        {input_contexto}
    </CONTEXTO>
    <PREGUNTA>
        {input_pregunta}
    </PREGUNTA>"

    Ignora el contexto si no es directamente relevante o si contradice lo que muestran los logs.

    Respuesta esperada (no incluyas lo que está entre [] en la respuesta, son solo aclaraciones):

    1. Detección de amenazas:
    [Sí/No, detalla las amenazas detectadas, si las hay]

    2. Nivel de riesgo:
    [Bajo - No hay riesgo ni amenazas detectadas,
    Medio - Hay actividades o posibles amenazas, pero no se detecta un uso malicioso claro,
    Alto - Se detectan amenazas claras siendo explotadas,
    Extremo - Amenazas críticas que requieren atención inmediata]

    3. Explicación técnica:
    [Análisis detallado de los logs basándote en el contexto proporcionado (si lo hay), incluyendo
    referencias a documentos relevantes (si hay contexto)]

    4. Cursos de Acción (COAs):
    [Recomendaciones específicas para mitigar el problema, incluyendo pasos a seguir y medidas
    preventivas. Proporciona medidas específicas adaptadas al caso; evita recomendaciones genéricas]

    Asegúrate de que la respuesta sea clara, basada en evidencias, y útil para la toma de decisiones
    de un analista de seguridad.
    """

    prompt = ChatPromptTemplate.from_template(template)
    llm = ChatOpenAI(model=DEF_EMBED_MODEL, temperature=0)
    logging.info("Modelo y prompt configurados.")
    return prompt, llm
```

Figura 4.8: Función de definición del prompt en el código

Además, se define de forma explícita la estructura de la respuesta, que debe incluir una evaluación de amenazas, una estimación del nivel de riesgo, una explicación técnica fundamentada y un conjunto de cursos de acción (COAs) detallados y adaptados al caso. Esta estructura permite estandarizar las salidas del modelo, facilitando su interpretación y validación por parte de analistas humanos, así como su integración en herramientas de seguimiento de incidentes. La implementación del prompt se ha realizado en código Python mediante plantillas reutilizables, lo que permite su adaptación y mantenimiento en entornos reales.

Ejemplo de Respuesta Generada

En este diseño se especifican claramente los datos que el modelo va a recibir (los logs, el contexto y la pregunta del usuario), así como la estructura exacta que debe tener la respuesta. Además, se le pide expresamente que fundamente sus análisis en los datos proporcionados, evitando con ello posibles alucinaciones o respuestas genéricas sin justificación.

Este enfoque ha demostrado ser eficaz. A continuación, se muestra un ejemplo realista de respuesta generada a partir de un conjunto de logs y una pregunta concreta:

> **Pregunta del usuario:** *¿Existe algún problema en estos logs? ¿Qué medidas deberían tomarse para mitigar el riesgo?*

Con esta pregunta, se espera que el sistema genere una respuesta estructurada como la de la siguiente Figura:

```
1. Detección de amenazas: Sí. Se ha identificado actividad sospechosa
relacionada con la técnica "T1078 - Cuentas Válidas" del marco MITRE ATT&CK.

2. Nivel de riesgo: Alto. Se detectó un intento de autenticación desde una
dirección IP sospechosa utilizando credenciales legítimas.

3. Explicación técnica:
   - Se ha registrado un intento de acceso desde una IP no habitual para este
   usuario.
   - La dirección IP está incluida en listas de amenazas conocidas.
   - La técnica utilizada sugiere un posible uso de credenciales
   comprometidas.

4. Cursos de Acción (COAs):
   - Bloquear la IP sospechosa en el firewall.
   - Restablecer credenciales para el usuario afectado.
   - Realizar una auditoría de accesos recientes para detectar posibles
   compromisos adicionales.
```

Figura 4.9: Ejemplo de respuesta generada por el sistema

Este tipo de respuesta refleja claramente el valor del diseño del prompt. No solo identifica con precisión la amenaza, sino que contextualiza el riesgo basándose en información externa, proporciona una explicación técnica comprensible y recomienda medidas concretas y accionables.

En otras palabras, convierte los datos crudos de los logs en conocimiento útil y entendible para la toma de decisiones.

5. Resultados

A lo largo de este apartado se presentan los principales resultados obtenidos tras la implementación del sistema. Se analizan tanto su comportamiento ante eventos de seguridad como la calidad de las respuestas generadas por el modelo, con especial atención a la clasificación de amenazas y la adecuación de las recomendaciones propuestas.

5.1. Escenarios de Validación

Con el objetivo de evaluar la capacidad del sistema desarrollado para detectar, contextualizar y responder ante incidentes de seguridad, se han planteado dos escenarios de validación. Ambos reproducen entornos realistas en los que se generan eventos de seguridad, los cuales son posteriormente analizados por el sistema RAG+LLM. Esta sección describe el desarrollo y finalidad de cada uno de estos casos.

5.1.1. Escenario 1: Detección de amenazas simuladas con CALDERA

Este escenario se desarrollará en una máquina virtual Windows, configurada como *endpoint* con agente de Wazuh. En ella, se ejecutará un escenario de emulación de adversarios utilizando la guía oficial de Wazuh y CALDERA [76].

CALDERA permitirá simular comportamientos de actores maliciosos reales, siguiendo el marco MITRE ATT&CK. Los ataques que emulará incluyen las siguientes técnicas:

- **T1197 - BITS Jobs:** Uso malicioso del servicio BITS de Windows para descargar o ejecutar archivos en segundo plano, facilitando la persistencia del atacante.
- **T1040 - Network Sniffing:** Captura de tráfico de red para obtener información sensible como credenciales o datos no cifrados.
- **T1021.001 - Remote Desktop Protocol:** Acceso remoto a sistemas mediante Remote Desktop Protocol, utilizado para movimientos laterales o control total del equipo.

Tras definir las reglas de detección en Wazuh (según la guía mencionada anteriormente), durante la simulación se registrarán los eventos clave, como modificaciones de cuentas de usuario, ejecución de tareas maliciosas y cambios en configuraciones del sistema. Estos logs serán procesados por el sistema RAG, que consultará la base de conocimiento estructurada con información táctica y técnica (matrices MITRE, documentos NIST, etc.).

LLaMA 3.1 clasificará cada evento según su nivel de criticidad, explicando las causas del incidente y proponiendo COAs adaptados al entorno y técnica empleada. Este escenario

permitirá comprobar la capacidad del sistema para interpretar logs generados artificialmente, validando su desempeño en entornos controlados y fácilmente reproducibles.

5.1.2. Escenario 2: Práctica de CTF de la asignatura SEGU

El segundo escenario se basa en la ejecución de una práctica de tipo *Capture The Flag* (CTF), enmarcada en la asignatura *Seguridad en Sistemas y Redes de Telecomunicación* (SEGU), donde se emula una intrusión por parte de un exempleado con acceso previo al entorno.

La práctica [4], se lleva a cabo sobre una imagen OVA con tres máquinas virtuales (máquina atacante, host accesible y servidor interno), configuradas mediante un script en Python. El usuario solo tiene acceso inicial a la máquina atacante, desde la cual debe escalar privilegios y pivotar lateralmente para acceder a otras partes de la red. El entorno, descrito en la Figura 5.1, incluye servicios como SSH, FTP y aplicaciones web con posibles vulnerabilidades explotables (inyección SQL, movimientos laterales, descifrado de credenciales, etc.).

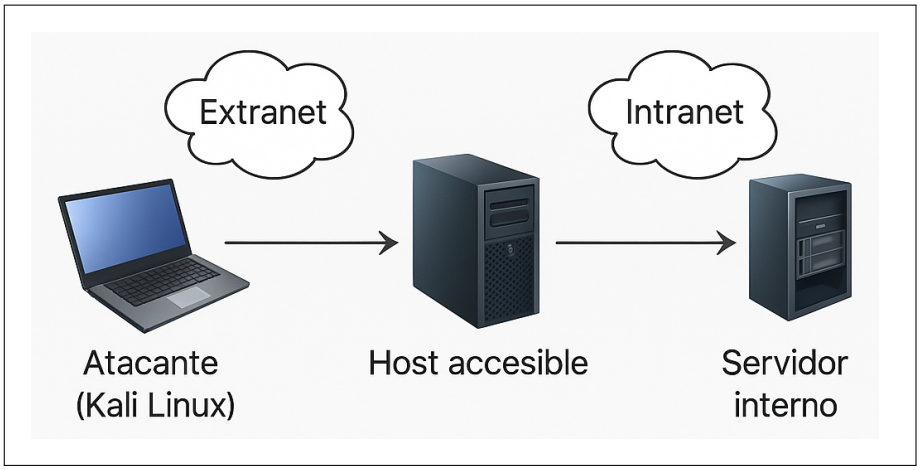

Figura 5.1: Esquema del entorno virtualizado en la práctica CTF.
Imagen obtenida de [4].

En este caso, los logs analizados corresponden a una única ejecución completa del ataque, realizada por el autor del TFG. Wazuh será configurado para recolectar eventos relevantes, en particular aquellos definidos por las reglas del motor OSSEC, como accesos remotos, modificación de archivos sensibles y ejecuciones sospechosas. Estas reglas están documentadas en el repositorio oficial de Wazuh [77].

El sistema RAG analizará estos logs consultando documentación técnica, catálogos de vulnerabilidades y descripciones de TTPs relacionadas con los pasos seguidos en la intrusión. Posteriormente, el modelo LLM asignará un nivel de criticidad a cada evento y propondrá medidas concretas para mitigar las amenazas detectadas, como el aislamiento de máquinas comprometidas, la revocación de credenciales o la reconfiguración de accesos remotos.

Este escenario pretende demostrar la capacidad del sistema no solo como herramienta de detección, sino también como apoyo a la formación en ciberseguridad, al facilitar la interpretación técnica de los incidentes y fomentar la toma de decisiones informadas.

5.2. Pruebas a realizar

Para validar el rendimiento del sistema propuesto y demostrar el impacto del enfoque RAG en el análisis de ciberamenazas, se ha diseñado un conjunto de pruebas que aplicaremos de forma sistemática a ambos casos de uso definidos en el apartado anterior.

Estas pruebas se realizarán en dos configuraciones del sistema, ambas con el mismo *prompt*:

- **LLM sin RAG**: el modelo de lenguaje opera exclusivamente con la información contenida en los logs, sin acceso a bases de conocimiento externas.
- **LLM con RAG**: el modelo utiliza, además de los logs, información contextual recuperada desde una base de conocimiento estructurada, mediante el sistema RAG.

El objetivo de esta doble evaluación es determinar, con precisión, el grado de mejora que introduce el uso de RAG en términos de precisión, contextualización y utilidad de las respuestas.

Métricas de evaluación

Las métricas de evaluación definidas son las siguientes:

Prueba	Objetivo	Métrica / Método
Precisión en la detección	Evaluar si el sistema identifica correctamente las amenazas reales presentes en los logs	Evaluación cualitativa
Clasificación por criticidad	Valorar si el sistema asigna el nivel de riesgo adecuado	Evaluación cualitativa
Calidad de los COAs	Medir la utilidad, claridad y aplicabilidad de las recomendaciones generadas	Escala de Likert (1–5)
Tiempo de respuesta	Cuantificar el tiempo necesario desde la entrada del log hasta la generación del informe	Evaluación cuantitativa (tiempo en segundos)
Nivel de contextualización	Evaluar si la respuesta integra información externa de forma coherente y útil	Evaluación cualitativa

Tabla 5.1: Métricas de evaluación

Tanto la métrica de *Precisión en la detección* como la de *Clasificación por criticidad* serán evaluadas en base al conocimiento base que ya se tiene de cada caso de uso. En el escenario de CALDERA, se conocen de antemano los ataques simulados, mientras que en el escenario de la práctica de CTF, al ser una práctica guiada, se conocen también los pasos y técnicas que se ejecutan.

Como se indica en la Tabla 5.1, para evaluar la calidad de los COAs generados se utilizará una escala de Likert [78] de cinco puntos, una herramienta ampliamente empleada para medir valoraciones cualitativas de forma estructurada. Esta escala permite puntuar cada recomendación según tres criterios fundamentales: claridad en la redacción, relevancia respecto a la amenaza detectada y aplicabilidad en un entorno real. Cada aspecto será valorado con una puntuación en un rango de 1 (muy deficiente) a 5 (excelente). La media de estas puntuaciones se empleará como indicador global de calidad del COA, lo que permitirá comparar objetivamente ambas configuraciones del sistema (con y sin RAG) en términos de utilidad práctica de las recomendaciones generadas.

El tiempo de respuesta se va a calcular como el intervalo entre la recepción del log y la generación del informe completo por parte del sistema. En ambos casos, los tiempos serán medidos mediante la función `time()` de Python bajo condiciones de prueba controladas y equivalentes para asegurar su comparabilidad.

Finalmente, para evaluar el nivel de contextualización, se va a utilizar la herramienta LangSmith, que permite observar los documentos concretos recuperados como contexto por el sistema RAG. El análisis se centrará en valorar la relevancia y adecuación de estos documentos con respecto al contenido de los logs analizados en cada caso de uso.

Todos los resultados que se obtendrán serán comparados entre las dos versiones del sistema para evaluar el valor añadido del enfoque RAG. El objetivo de esta evaluación es justificar, con datos, la elección de un sistema RAG frente a un modelo generativo aislado.

5.3. Resultados obtenidos

A partir de la ejecución de los casos de prueba definidos en el apartado anterior, se han obtenido resultados diferenciados entre las dos configuraciones del sistema: el modelo de lenguaje operando de forma aislada (LLM sin RAG) y el sistema enriquecido con recuperación aumentada de información (LLM con RAG). A continuación, se presentan los resultados obtenidos en cada una de las métricas evaluadas, organizados por caso de uso. Los resultados en bruto obtenidos en ambos caso de prueba se encuentran en el *Anexo C: Resultados obtenidos con el Código*.

5.3.1. Caso de prueba: Simulación de ataques con CALDERA

En este escenario, el sistema ha analizado un conjunto de logs generados mediante una simulación ejecutada con la herramienta CALDERA. El objetivo era evaluar la capacidad del sistema para identificar configuraciones inseguras y comportamientos asociados a técnicas de intrusión.

Precisión en la detección

Como se puede observar en las siguiente figuras, ambos enfoques, tanto el modelo LLM sin RAG como el sistema con RAG, detectan la presencia de problemas en los logs analizados. Sin embargo, la naturaleza y profundidad de las detecciones difieren notablemente entre ambas configuraciones. En el caso sin RAG (Figura 5.2), el modelo identifica configuraciones inseguras de forma general, sin aportar detalles técnicos específicos ni relacionarlos con patrones conocidos de comportamiento malicioso.

```
================================ RESULTADO SIN RAG ================================

1. Detección de amenazas:
    Sí, hay problemas en los logs. Se detectan varias configuraciones inseguras
    que pueden comprometer la seguridad del sistema.
```

Figura 5.2: C1 (sin RAG) - Detección de amenazas

Por otro lado, la configuración con RAG (Figura 5.3) proporciona una descripción más detallada del incidente, incluyendo referencias explícitas a técnicas del marco MITRE ATT&CK. En este análisis se identifican, entre otras, técnicas como File/Path Exclusions, Custom Cryptographic Protocol, Clear Linux or Mac System Logs y Process Injection, lo que indica un mayor nivel de contextualización de la amenaza. Esta diferencia en la especificidad del análisis sugiere que la integración de información estructurada mediante RAG permite al sistema vincular los eventos observados con tácticas reconocidas, aportando un mayor grado de detalle y trazabilidad en la detección.

```
================================ RESULTADO CON RAG ================================

1. Detección de amenazas:
    Sí, hay problemas en estos logs que indican posibles actividades maliciosas.
    * Se menciona la técnica ATT&CK "File/Path Exclusions", lo que sugiere que un
    atacante puede estar intentando ocultar sus archivos y evadir las
    herramientas de seguridad.
    * Se describe la técnica ATT&CK "Custom Cryptographic Protocol", lo que
    podría indicar que un atacante está utilizando un protocolo criptográfico
    personalizado para ocultar su tráfico de control remoto.
    * Se menciona la técnica ATT&CK "Clear Linux or Mac System Logs", lo que
    sugiere que un atacante puede estar intentando borrar los registros del
    sistema para evitar ser detectado.
    * Se describe la técnica ATT&CK "Process Injection", lo que podría indicar
    que un atacante está inyectando código en procesos legítimos para evadir las
    herramientas de seguridad y obtener privilegios elevados.
```

Figura 5.3: C1 (con RAG) - Detección de amenazas

Cabe destacar que, en ambos casos, las técnicas identificadas no coinciden con las previstas en el diseño del caso de prueba, lo que sugiere cierta desconexión entre los eventos registrados en los logs y las acciones efectivamente ejecutadas durante la simulación. Este aspecto se examina con mayor detalle en la sección siguiente.rtando un mayor grado de detalle y trazabilidad en la detección.

Clasificación por criticidad

En cuanto a la valoración del nivel de riesgo, en ambos enfoques el sistema clasifica los eventos como de riesgo alto, lo que refleja un reconocimiento de la gravedad de las amenazas detectadas. No obstante, los argumentos técnicos que respaldan esta evaluación presentan diferencias significativas.

En los resultado obtenidos sin RAG (Figura 5.4), la asignación del nivel de riesgo se fundamenta principalmente en la observación directa de configuraciones inseguras dentro del

sistema. Entre los elementos identificados destacan la activación del servicio Link-Layer Topology Discovery Mapper, el uso inapropiado del modo de aprobación de administrador (UAC) y la presencia de software malicioso instalado sin validación. Aunque estas detecciones son relevantes, el análisis permanece limitado al contenido literal de los logs y no da suficiente información que permitan contextualizar la amenaza de forma más precisa.

```
========================== RESULTADO SIN RAG ==========================

2. Nivel de riesgo:
    Alto. La presencia de varias amenazas y la explotación de
    vulnerabilidades sugieren un nivel de riesgo alto.

3. Explicación técnica:
    Se han encontrado varios problemas en los logs:
    * El servicio "Link-Layer Topology Discovery Mapper (lltdsvc)"
    está habilitado, lo que podría permitir la explotación de
    vulnerabilidades.
    * Se detectan intentos de acceso no autorizados a servicios y
    componentes del sistema operativo, como el servicio "User Account
    Control: Admin Approval Mode for the Built-in Administrator account".
    * Se observa la instalación de software malicioso en el sistema.
```

Figura 5.4: C1 (sin RAG) - Nivel de riesgo y Explicación técnica

Por otro lado, la configuración con RAG, visible en la Figura **??**, permite una clasificación más justificada y profunda del riesgo. En este caso, el sistema identifica diversas técnicas descritas en el marco MITRE ATT&CK, como Process Injection, File/Path Exclusions, Custom Cryptographic Protocol y Clear Linux or Mac System Logs. Estas técnicas apuntan a comportamientos típicos de adversarios con un nivel avanzado de sofisticación operativa, lo que refuerza la validez de la clasificación de riesgo asignada.

```
========================== RESULTADO CON RAG ==========================

2. Nivel de riesgo:
    Alto. Estos logs sugieren que hay una actividad maliciosa en curso,
    pero no es claro si se trata de un ataque avanzado o simplemente de
    una exploración de vulnerabilidades. El nivel de riesgo es alto porque
    la detección de amenazas es significativa y requiere atención
    inmediata. Se detecta una posible vulnerabilidad en la configuración
    de seguridad del sistema operativo Windows, específicamente en la
    configuración de cuentas de usuario y permisos.
```

Figura 5.5: C1 (con RAG) - Nivel de riesgo

Además, el modelo contextualiza la amenaza incorporando elementos adicionales como la configuración de permisos en cuentas administrativas y de usuario, sugiriendo la existencia de vectores de ataque potenciales derivados de una gestión deficiente de privilegios. Esta capacidad de relacionar vulnerabilidades técnicas con escenarios reales de explotación aporta una dimensión estratégica al análisis de criticidad.

Es importante destacar una limitación mencionada previamente compartida por ambas configuraciones del sistema. Al contrastar las técnicas identificadas durante el análisis de los logs con las efectivamente emuladas en la simulación llevada a cabo con CALDERA, se observa una clara discrepancia. La guía utilizada para construir este caso de uso establece que los ataques aplicados corresponden a técnicas concretas como T1197 (BITS Jobs), T1040 (Network Sniffing) y T1021.001 (Remote Desktop Protocol), entre otras en sistemas Linux. No obstante, ni el modelo sin RAG ni el modelo con RAG lograron identificar estas técnicas de forma explícita en sus análisis. En su lugar, se asociaron los eventos a otras técnicas distintas del marco MITRE ATT&CK que no formaban parte del perfil adversario configurado. Esta divergencia sugiere una limitación en la capacidad del sistema para correlacionar los eventos observados con las amenazas reales ejecutadas, lo cual afecta directamente a la precisión en la valoración del riesgo. Esta observación resulta relevante para la valoración global del sistema y será abordada con mayor detalle en el apartado de conclusiones, donde se discutirán posibles causas y estrategias de mejora.

```
========================== RESULTADO CON RAG ==========================

3. Explicación técnica:
    * La técnica "File/Path Exclusions" implica que el atacante
    está utilizando una lista de exclusión para evitar que las
    herramientas de seguridad detecten sus archivos.
    * La técnica "Custom Cryptographic Protocol" sugiere que el atacante
    está utilizando un protocolo criptográfico personalizado para ocultar
    su tráfico de control remoto.
    * La técnica "Clear Linux or Mac System Logs" implica que el atacante
    está borrando los registros del sistema para evitar ser detectado.
    * La técnica "Process Injection" sugiere que el atacante
    está inyectando código en procesos legítimos para evadir las
    herramientas de seguridad y obtener privilegios elevados.
    * Los logs indican que hay una cuenta de administrador con el nombre
    "Administrador" que tiene permisos elevados en el sistema. Esto puede
    ser un problema si no se ha configurado correctamente, ya que permite
    a cualquier persona con acceso al sistema realizar cambios importantes
    sin necesidad de autenticación adicional.
    * Además, los logs también muestran que hay una cuenta de usuario con
    el nombre "Usuario" que tiene permisos de escritura en la carpeta de
    documentos. Esto puede ser un problema si no se ha configurado
    correctamente, ya que permite a cualquier persona con acceso al
    sistema modificar archivos importantes sin necesidad de autenticación
    adicional.
    * Es importante mencionar que estos problemas pueden ser causados por
    una configuración incorrecta o por una falta de actualización de
    seguridad en el sistema operativo.
```

Figura 5.6: C1 (con RAG) - Explicación técnica

Calidad de los Cursos de Acción

Respecto a las recomendaciones generadas tras el análisis, en la configuración sin RAG, el modelo ofrece un conjunto de acciones centradas en la corrección directa de los elementos detectados en los registros. Las medidas recomendadas incluyen la desactivación de servicios específicos, la revisión de configuraciones de control de cuentas de usuario (UAC) y la ejecución de análisis de seguridad adicionales. Estas propuestas, si bien son pertinentes y de fácil aplicación, se limitan a una intervención localizada y reactiva, sin considerar el contexto operativo ni la proyección a largo plazo de las medidas.

```
============================== RESULTADO SIN RAG ==============================

4. Cursos de Acción (COAs):
    Para mitigar las amenazas detectadas, se recomienda lo siguiente:
    - Desactivar el servicio "Link-Layer Topology Discovery Mapper (lltdsvc)":
    Esto ayudará a prevenir la explotación de vulnerabilidades en este servicio.
    - Revisar y actualizar los permisos del servicio "User Account Control: Admin
    Approval Mode for the Built-in Administrator account": Esto ayudará a
    prevenir el acceso no autorizado a servicios y componentes del sistema
    operativo.
    - Realizar un análisis de seguridad exhaustivo: Esto ayudará a identificar y
    mitigar cualquier otra amenaza potencial en el sistema.
    - Implementar medidas de seguridad adicionales: Esto podría incluir la
    implementación de firewalls, antivirus y sistemas de detección de intrusos
    para prevenir futuras amenazas.
```

Figura 5.7: C1 (sin RAG) - Cursos de Acción

Por el contrario, al incorporar el sistema RAG, las recomendaciones adquieren un carácter más estratégico y global. El modelo propone un conjunto de acciones más amplio, que abarca aspectos tanto técnicos como organizativos. Entre las medidas destacadas se incluyen el uso de autenticación multifactor, la implementación del principio de mínimo privilegio (PoLP), la auditoría de configuraciones sensibles, la utilización de herramientas EDR avanzadas, el bloqueo de protocolos sospechosos mediante DPI, y la gestión proactiva de los registros del sistema mediante servidores centralizados. Estas acciones reflejan una comprensión más madura del ciclo de vida de la ciberdefensa y se alinean con buenas prácticas ampliamente reconocidas en el sector.

```
============================== RESULTADO CON RAG ==============================

4. Cursos de Acción (COAs):
    - Renombrar o deshabilitar la cuenta "Administrador" predeterminada para
    evitar accesos no controlados.
    - Aplicar autenticación multifactor (MFA) en todas las cuentas con
    privilegios elevados.
    - Limitar los permisos de escritura en carpetas sensibles, como \Documentos",
    solo a usuarios autorizados.
    - Aplicar el principio de mínimo privilegio (PoLP) para todos los usuarios y
    servicios.
    - Auditar y restringir las exclusiones de archivos y rutas en antivirus y
    soluciones EDR.
    - Utilizar EDR con capacidades de detección de inyección de procesos y
    técnicas de evasión avanzadas.
    - Redirigir los logs a un servidor centralizado para impedir su borrado o
    manipulación local.
    - Detectar y alertar sobre el borrado o vaciado de archivos de logs, con
    herramientas SIEM o scripts automatizados.
    - Bloquear protocolos personalizados o no autorizados en firewall y proxy, y
    aplicar inspección profunda (DPI).
    - Aplicar actualizaciones de seguridad de forma regular y automatizada,
    incluyendo sistema operativo y software clave.
```

Figura 5.8: C1 (con RAG) - Cursos de Acción

Para reflejar con claridad estas diferencias cualitativas emplearemos la escala de Likert. En el caso del modelo sin RAG, las recomendaciones son comprensibles y ejecutables, lo

que justifica una puntuación aceptable en claridad y aplicabilidad (3.0 en ambos casos). Sin embargo, su relevancia es limitada (2.5), ya que las acciones propuestas carecen de un vínculo explícito con marcos normativos o con las amenazas detectadas de forma estructurada. Frente a esto, la configuración con RAG obtiene valoraciones notablemente superiores en los tres criterios, destacando especialmente en relevancia (4.7), debido a la alineación directa entre los COAs propuestos y las técnicas identificadas del marco MITRE ATT&CK. La claridad (4.5) y la aplicabilidad (4.5) también alcanzan niveles elevados, reflejando tanto la coherencia del lenguaje utilizado como la viabilidad de las medidas en entornos operativos reales. Estas puntuaciones, reflejadas en la Tabla 5.2 refuerzan la conclusión de que la integración de RAG mejora sustancialmente la utilidad práctica de las recomendaciones generadas.

Caso de uso	Configuración	Claridad	Relevancia	Aplicabilidad	Punt. media
Logs_C1	LLM sin RAG	3.0	2.5	3.0	**3.2**
	LLM con RAG	4.5	4.7	4.5	**4.6**

Tabla 5.2: Evaluación en escala Likert

No obstante, esta mejora cualitativa en los COAs debe analizarse con cuidado. Tal como se ha señalado en la sección anterior, las técnicas identificadas por el sistema durante el análisis no se corresponden con las que fueron realmente emuladas mediante CALDERA. Esta falta de alineación afecta directamente a la pertinencia de las acciones propuestas, ya que si bien las recomendaciones son técnicamente válidas y coherentes con las técnicas que el modelo infiere, no responden necesariamente a los vectores de ataque efectivamente desplegados en el entorno. Esto reduce la aplicabilidad contextual real de los COAs generados, particularmente en entornos donde se espera una correspondencia directa entre detección, análisis y respuesta. Esta limitación, común a ambas configuraciones (con y sin RAG), será tratada en profundidad en el apartado de conclusiones, donde se propondrán estrategias de mejora para reforzar la fiabilidad del sistema.

Tiempo de respuesta

En relación con la eficiencia del sistema, la configuración sin RAG presentó un tiempo medio de 14,96 segundos, mientras que la configuración con RAG obtuvo una media inferior de 9,37 segundos. Este resultado, observado de forma consistente en todas las ejecuciones, sugiere que la incorporación del sistema de recuperación de información no solo no penaliza el rendimiento, sino que puede contribuir a agilizar la generación de respuestas. Esta diferencia es muy relevante de cara a la valoración global del sistema.

Nivel de contextualización

Por último, se ha analizado el grado de contextualización alcanzado durante la ejecución del caso de prueba. En la Figura 5.9, obtenida empleando la herramienta LangSmith, se observa cómo el sistema recuperó principalmente fragmentos de la norma NIST 800-53, relacionados con controles de seguridad como protección de fronteras, restricciones de comunicación, validación de atributos y políticas de actualización. Aunque estos documentos aportan una base normativa útil, su relación con los eventos concretos observados en los logs es más general que específica.

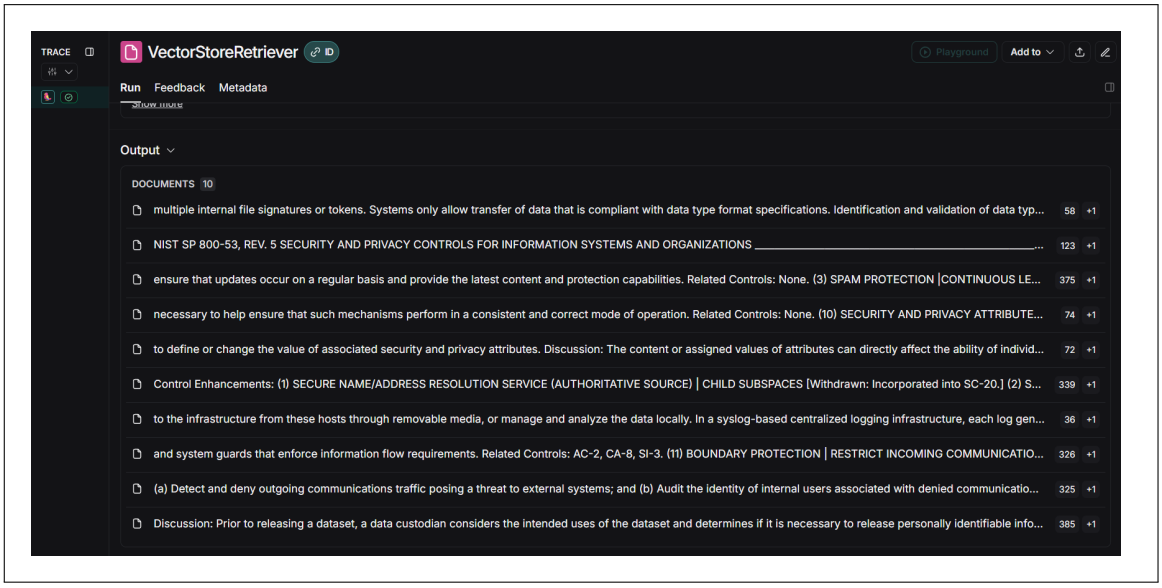

Figura 5.9: C1 - Contexto extraído

Asimismo, el sistema también incorporó técnicas del marco MITRE ATT&CK como parte de su contextualización. No obstante, tal y como se ha expuesto en apartados anteriores, estas técnicas no coinciden con las que fueron emuladas realmente en la simulación con CALDERA. Esta discrepancia no implica necesariamente un error del sistema, ya que es posible que los logs generados contengan patrones u observaciones que se asemejen a otras técnicas registradas en el marco ATT&CK. Sin embargo, esta diferencia sí afecta al alineamiento entre el contexto recuperado y el escenario diseñado, lo que reduce en parte la efectividad del proceso de enriquecimiento semántico.

En resumen, los resultados obtenidos en este primer caso de prueba permiten observar diferencias claras entre ambas configuraciones del sistema. Si bien la integración del componente RAG ha demostrado aportar mejoras en términos de contextualización, calidad de las recomendaciones y tiempo de respuesta, también se han identificado limitaciones comunes en cuanto a la correlación precisa entre las técnicas detectadas y las acciones realmente ejecutadas. Estas observaciones servirán de base para el análisis comparativo con otros escenarios y se desarrollarán en profundidad en el apartado de conclusiones.

5.3.2. Caso de prueba: Práctica CTF

Este segundo escenario corresponde al análisis de registros generados en el transcurso de una práctica académica tipo Capture The Flag (CTF). A diferencia del caso anterior, los eventos aquí son más variados, menos estructurados y abarcan múltiples etapas de un ataque realista, lo que plantea un desafío adicional para el sistema en términos de detección, contextualización y respuesta.

Precisión en la detección

Ambas configuraciones del sistema, tanto con como sin RAG, son capaces de identificar la existencia de amenazas relevantes. En la configuración sin RAG (Figura 5.10), el modelo detecta varios indicadores de posible compromiso, como cambios en el checksum de integridad, incorporación de archivos no autorizados, uso de comandos con privilegios

mediante sudo y modificaciones en los puertos escuchados. Estas detecciones reflejan una capacidad razonable para identificar anomalías técnicas, aunque se presentan de forma relativamente genérica y sin referencias explícitas a tácticas o técnicas reconocidas.

```
============================== RESULTADO SIN RAG ==============================

1. Detección de amenazas:
   Sí, se detectan varias amenazas en los logs proporcionados.
   - La primera amenaza es el cambio en el checksum de integridad del sistema, lo
   que puede indicar una modificación no autorizada en el sistema.
   - Otra amenaza es la adición de archivos al sistema sin autorización, lo que
   puede ser un intento de introducir malware o código malicioso.
   - También se detecta el uso de `sudo` para ejecutar comandos con privilegios
   root, lo que podría ser una vulnerabilidad si no está configurado
   correctamente.
   - Además, hay cambios en los puertos escuchados por el sistema, lo que puede
   indicar la instalación de servicios o software sin autorización.
```

Figura 5.10: C2 (sin RAG) - Detección de amenazas

Por otro lado, la configuración con RAG (Figura 5.11) ofrece una caracterización más exhaustiva de las actividades observadas. El modelo no solo identifica los mismos elementos que en la versión sin RAG, sino que añade detalles como el acceso no autorizado a archivos sensibles del sistema (por ejemplo, el archivo de la base de datos NTDS de Active Directory), la creación de copias de seguridad sospechosas y el uso de herramientas como procdump para obtener volcados de memoria del proceso LSASS. Estas observaciones coinciden con técnicas habitualmente asociadas a ataques de escalada de privilegios y exfiltración de credenciales.

```
============================== RESULTADO CON RAG ==============================

1. Detección de amenazas:
   Sí, se detectan varias actividades sospechosas que pueden indicar una posible
   amenaza en el sistema. Estas incluyen:
   - Acceso no autorizado a archivos y directorios importantes, como
   `%SystemRoot%\\NTDS\\Ntds.dit`.
   - Creación de copias del archivo NTDS (NTDS.dit) en diferentes ubicaciones.
   - Uso de herramientas como `procdump` para dumping la memoria del proceso
   LSASS.
```

Figura 5.11: C2 (con RAG) - Detección de amenazas

Además, la configuración con RAG contextualiza mejor las acciones observadas, vinculándolas con patrones de ataque reconocidos, lo que facilita una interpretación más clara y estructurada de la amenaza. Esta mejora cualitativa en la precisión del análisis refuerza la utilidad del componente RAG para escenarios en los que la detección depende de una correcta interpretación semántica de eventos complejos.

Clasificación por criticidad

La evaluación del nivel de riesgo en los registros del escenario CTF muestra que ambas configuraciones del sistema coinciden en clasificar las actividades observadas como amena-

zas de riesgo alto. Sin embargo, al igual que en otros apartados, la argumentación técnica que sustenta esta clasificación difiere entre las dos configuraciones.

En el caso sin RAG, visible en la Figura 5.12, la asignación de criticidad se basa en la detección de eventos anómalos que podrían indicar una intrusión o una manipulación no autorizada del sistema. Entre los indicadores identificados destacan el cambio en los checksums de integridad del sistema, la adición no autorizada de archivos, el uso de comandos privilegiados mediante sudo, y la modificación de los puertos en escucha. Estas señales son relevantes y justifican una alerta de riesgo, pero la explicación permanece anclada a los síntomas observables, sin un análisis más profundo de su posible impacto operativo o su conexión con patrones de ataque conocidos.

```
============================== RESULTADO SIN RAG ==============================

2. Nivel de riesgo:
    Alto. Se detectan varias actividades sospechosas y posibles amenazas que
    requieren atención inmediata para evitar daños al sistema o a la seguridad en
    general.

3. Explicación técnica:
    Los logs muestran una serie de eventos que pueden indicar una posible
    intrusión o manipulación del sistema. El cambio en el checksum de integridad
    sugiere que alguien ha modificado el sistema, lo que podría ser un intento de
    ocultar malware o código malicioso.
    La adición de archivos sin autorización es otro indicio preocupante, ya que
    puede permitir la instalación de software malicioso. El uso de `sudo` para
    ejecutar comandos con privilegios root también es una señal de advertencia,
    especialmente si no está configurado correctamente, lo que podría dar acceso
    a un atacante.
    Los cambios en los puertos escuchados pueden indicar la instalación de
    servicios o software sin autorización, lo que puede comprometer la seguridad
    del sistema.
```

Figura 5.12: C2 (sin RAG) - Nivel de riesgo y Explicación técnica

Por el contrario, la configuración con RAG de la Figura 5.14, permite una evaluación más detallada y estratégica del nivel de amenaza. El sistema interpreta los eventos registrados no solo como anomalías técnicas, sino como pasos en un posible proceso de escalada de privilegios y exfiltración de datos. Se destaca el acceso no autorizado al archivo NTDS.dit, que almacena información crítica sobre cuentas de usuario en entornos Active Directory, y el uso de herramientas como procdump para obtener volcados de la memoria del proceso LSASS, una técnica ampliamente reconocida para la obtención de credenciales. La creación de copias del archivo NTDS en ubicaciones alternativas refuerza la hipótesis de persistencia y preparación para la exfiltración.

```
============================== RESULTADO CON RAG ==============================

2. Nivel de riesgo:
    Alto. La presencia de estas actividades sugiere que alguien está intentando
    acceder o copiar información sensible, lo que puede indicar una amenaza
    significativa.
```

Figura 5.13: C2 (con RAG) - Nivel de riesgo

```
================================ RESULTADO CON RAG ================================

3. Explicación técnica:
    Los logs muestran varias acciones sospechosas relacionadas con el acceso y la
    manipulación de archivos importantes en el sistema, incluyendo el archivo
    NTDS (NTDS.dit), que contiene información sensible sobre la red y los
    usuarios. El uso de herramientas como `procdump` para dumping la memoria del
    proceso LSASS también puede ser una señal de intento de acceso no autorizado
    a credenciales.

   - **Acceso no autorizado:** Se detecta el acceso no autorizado a archivos
   importantes, lo que podría indicar un intento de exfiltración de datos o
   explotación de vulnerabilidades.
   - **Creación de copias del archivo NTDS:** La creación de copias del archivo
   NTDS en diferentes ubicaciones puede ser una estrategia para mantener acceso a
   la información incluso si se detecta el acceso no autorizado original.
```

Figura 5.14: C2 (con RAG) - Explicación técnica

Estas evidencias aportadas por el sistema con RAG permiten entender la amenaza en términos operativos y no solo técnicos, vinculándola a un posible compromiso grave de la infraestructura interna. Esta capacidad de contextualización fortalece la justificación del nivel de riesgo asignado, consolidando el análisis como parte de un ciclo más completo de detección y respuesta.

Calidad de los Cursos de Acción

```
================================ RESULTADO SIN RAG ================================

4. Cursos de Acción (COAs):
    Para remediar la situación, se recomienda:
   - Realizar un análisis exhaustivo del sistema: Se debe realizar un análisis
   detallado del sistema para identificar cualquier posible amenaza o daño.
   - Revisar y actualizar las configuraciones de seguridad: Se deben revisar y
   actualizar las configuraciones de seguridad para asegurarse de que el sistema
   esté protegido contra posibles amenazas.
   - Monitorear el sistema: Se debe monitorear el sistema continuamente para
   detectar cualquier posible amenaza o daño.
   - Realizar un respaldo del sistema: Se debe realizar un respaldo del sistema
   para asegurarse de que se pueda restaurar el sistema en caso de una intrusión
   o daño significativo.
```

Figura 5.15: C2 (sin RAG) - Cursos de Acción

En lo relativo a las recomendaciones generadas por el sistema tras el análisis de los registros del caso de prueba CTF, se aprecian diferencias notables entre ambos enfoques. En la configuración sin RAG (Figura 5.15), los cursos de acción propuestos se centran en medidas generales y de carácter preventivo. Se sugiere realizar un análisis exhaustivo del sistema, revisar las configuraciones de seguridad, establecer mecanismos de monitorización continua y garantizar copias de respaldo. Aunque estas medidas son válidas desde un punto de vista operativo básico, carecen de una vinculación directa con los eventos específicos detectados, lo que limita su efectividad en el contexto concreto de la amenaza.

```
============================== RESULTADO CON RAG ==============================

4. Cursos de Acción (COAs):
   - Restringir el acceso al archivo NTDS.dit y a las carpetas del sistema
   (%SystemRoot%\NTDS) solo a cuentas y servicios estrictamente necesarios.
   - Bloquear y auditar el uso de herramientas como procdump, mimikatz,
   comsvcs.dll, y otros binarios usados para volcado de memoria.
   - Monitorizar en tiempo real los accesos y copias del archivo NTDS.dit,
   generando alertas ante acciones inusuales.
   - Redirigir los logs de seguridad a un servidor central (SIEM) para evitar su
   manipulación local y facilitar su análisis.
   - Revisar y limitar el número de cuentas con permisos administrativos,
   aplicando PoLP (Principio de Mínimo Privilegio).
   - Implementar alertas sobre creación de archivos sospechosos en ubicaciones
   críticas, como copias del NTDS.dit fuera de su ruta normal.
   - Aplicar parches de seguridad del sistema operativo y servicios de directorio
   (Active Directory) de forma regular y automatizada.
   - Realizar auditorías regulares del uso de herramientas administrativas y
   accesos privilegiados, con revisión cruzada por parte del equipo de seguridad.
```

Figura 5.16: C2 (con RAG) - Cursos de Acción

En cambio, en los resultados del sistema con RAG (Figura 5.16), se ofrece un conjunto de recomendaciones mucho más orientado al incidente observado y a sus implicaciones técnicas. Las acciones propuestas abordan con precisión aspectos clave como la protección del archivo *NTDS.dit*, el control del uso de herramientas de volcado de memoria como *procdump*, la centralización de logs en servidores SIEM para evitar manipulaciones, la aplicación del principio de mínimo privilegio, y la auditoría proactiva de accesos administrativos. Estas recomendaciones no solo muestran una mayor comprensión de la amenaza, sino que también están alineadas con buenas prácticas en entornos corporativos y marcos de referencia como NIST o MITRE.

La evaluación mediante escala Likert, reflejada en la Tabla 5.3 recoge estas diferencias cualitativas de forma estructurada. En la configuración sin RAG, las puntuaciones reflejan una aceptable claridad (3.0) y aplicabilidad (3.0), pero una baja relevancia (2.3), al no establecer una conexión directa entre las medidas propuestas y las amenazas identificadas. En cambio, la configuración con RAG obtiene valoraciones claramente superiores en los tres criterios. La claridad (4.7) y la aplicabilidad (4.5) reflejan tanto la redacción comprensible como la viabilidad de las acciones sugeridas. La relevancia (4.8) es especialmente destacable, dada la correspondencia entre las recomendaciones y los eventos observados en los registros, así como la integración de buenas prácticas reconocidas.

Caso de uso	Configuración	Claridad	Relevancia	Aplicabilidad	Punt. media
Logs_C2	LLM sin RAG	3.0	2.3	3.0	**2.8**
	LLM con RAG	4.7	4.8	4.5	**4.7**

Tabla 5.3: Evaluación en escala Likert

Tiempo de respuesta

En este segundo escenario, los tiempos de respuesta registrados refuerzan la tendencia observada en el caso anterior. La configuración sin RAG presentó un tiempo medio de

15,75 segundos, mientras que la configuración con RAG obtuvo un tiempo inferior, con una media de 10,41 segundos. Esta diferencia resulta especialmente relevante si se considera que la arquitectura RAG incorpora una fase adicional de recuperación de información, lo que en principio podría hacer prever un aumento en el tiempo de inferencia.

El resultado, no obstante, sugiere que la presencia de contexto externo puede facilitar el razonamiento del modelo, al reducir la necesidad de generar respuestas partiendo exclusivamente del conocimiento implícito en sus pesos. Esta hipótesis, si bien requiere ser contrastada con mayor profundidad, apunta a una ventaja operativa del enfoque RAG en términos de eficiencia, más allá de la mejora cualitativa en la calidad del análisis. Este comportamiento se analizará con mayor detenimiento en el apartado de conclusiones.

Nivel de contextualización

En este caso, cuando observamos el pipeline mediante LangSmith, se muestra cómo la versión del sistema con RAG accedió a un conjunto de documentos estructurados provenientes de marcos de conocimiento especializados, entre ellos MITRE ATT&CK y CAPEC. Tal y como se observa en la Figura 5.17, los contenidos recuperados incluyen técnicas como el acceso o copia del archivo NTDS.dit, la extracción de credenciales mediante ataques a la memoria del proceso LSASS, y múltiples patrones de ataque descritos en CAPEC relacionados con la elevación de privilegios y la explotación de debilidades en la validación de entradas.

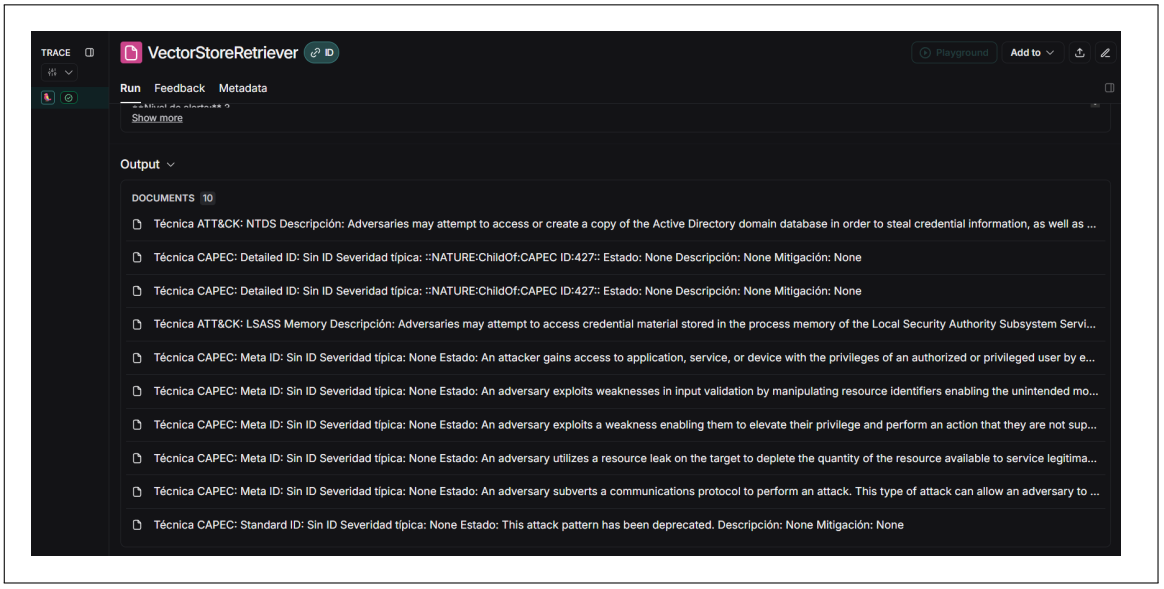

Figura 5.17: C2 - Contexto extraído

Bajo estas condiciones, el contenido contextual recuperado se alinea de forma bastante adecuada con los eventos observados en los registros. A diferencia del caso de prueba anterior, donde existía una clara disociación entre las técnicas recuperadas y las efectivamente ejecutadas, aquí el sistema RAG logra vincular su análisis con acciones altamente representativas del escenario: acceso a credenciales, movimientos laterales, y técnicas de post-explotación propias de un entorno de escalada de privilegios dentro de una red empresarial.

La aparición reiterada de referencias al archivo NTDS.dit, a herramientas como procdump,

y a procesos críticos como LSASS, demuestra que el sistema fue capaz de identificar los conceptos clave del incidente y recuperar material técnico directamente relacionado con ellos. Asimismo, las entradas de CAPEC complementan la contextualización ofreciendo una visión más amplia de los patrones de ataque subyacentes.

La diversidad de entradas consultadas, tanto por tipo de técnica como por origen (ATT&CK, CAPEC), evidencia que el modelo no opera únicamente en función de coincidencias literales, sino que se apoya en un proceso de recuperación semántica que le permite contextualizar las amenazas detectadas dentro de un marco de conocimiento estructurado.

Sin embargo, se observa también que varios de los documentos provenientes de CAPEC carecen de contenido útil, ya que aparecen sin descripciones ni directrices de mitigación. Esta circunstancia limita parcialmente el valor añadido del proceso de recuperación, ya que aunque el sistema accede a fuentes pertinentes, no siempre logra filtrar entradas con información aprovechable para enriquecer el análisis.

En conjunto, los resultados del escenario CTF refuerzan la ventaja del enfoque RAG frente al modelo base en términos de precisión, contextualización y calidad de respuesta, aunque también revelan oportunidades de mejora en la selección de fuentes documentales. Estas observaciones se retomarán en el apartado de conclusiones para identificar líneas futuras de optimización del sistema.

6. Conclusiones y líneas futuras

6.1. Conclusiones

Este Trabajo de Fin de Grado ha tenido como objetivo principal el diseño e implementación de un sistema de análisis de ciberamenazas basado en el enfoque de Generación Aumentada por Recuperación, integrando un modelo de lenguaje de última generación, LLaMA 3.1, con un sistema de Monitorización y Gestión de Eventos de Seguridad, Wazuh. A través de esta integración, se ha buscado no solo automatizar el análisis de logs generados en entornos de ciberseguridad, sino también ofrecer respuestas contextualizadas y accionables en forma de Cursos de Acción ante los incidentes detectados.

Durante el desarrollo del sistema, se ha comprobado la viabilidad tanto técnica como conceptual de esta arquitectura, evidenciando cómo los modelos de lenguaje pueden asumir un rol activo en procesos de detección, análisis y respuesta dentro de entornos de ciberdefensa. Para ello, se partió de una arquitectura donde los eventos son generados mediante simulaciones controladas, recogidos por Wazuh, y posteriormente analizados por un modelo LLM, al que se dota de contexto adicional mediante un sistema RAG conectado a una base vectorizada de documentos técnicos.

Uno de los aportes más relevantes del trabajo ha sido comprobar que la integración del sistema RAG mejora la precisión, profundidad y coherencia de los análisis realizados por el modelo. En particular, el uso de fuentes como el marco MITRE ATT&CK, el NIST SP 800-53 y la base de patrones CAPEC ha permitido enriquecer semánticamente las respuestas generadas, alineándolas con estándares reconocidos y reduciendo significativamente la aparición de alucinaciones típicas en modelos LLM no especializados.

Los resultados experimentales obtenidos en los dos escenarios de validación han evidenciado diferencias sustanciales entre la configuración base y la configuración con RAG. En ambos casos, el sistema con RAG logró:

- Detectar amenazas con mayor especificidad y contexto.
- Clasificar riesgos de forma más argumentada y vinculada a comportamientos reconocidos de actores maliciosos.
- Proponer COAs más precisos, relevantes y alineados con marcos normativos y buenas prácticas.
- Proponer COAs más precisos, relevantes y alineados con marcos normativos y buenas prácticas.
- Reducir el tiempo medio de respuesta, lo que sugiere que el contexto adicional facilita el razonamiento del modelo.

- Recuperar información técnica pertinente y específica, mejorando la trazabilidad y explicabilidad del análisis.

No obstante, uno de los principales hallazgos críticos del proyecto ha sido la dificultad del sistema para correlacionar con precisión los eventos registrados en los logs con las técnicas que realmente fueron emuladas durante la ejecución de los escenarios. Esta limitación, presente tanto en la versión con RAG como sin RAG, tiene implicaciones relevantes para la confiabilidad del sistema en entornos donde la trazabilidad entre actividad ejecutada y análisis automático es fundamental.

En el caso de la simulación con CALDERA, por ejemplo, se emularon técnicas específicas como T1197, T1040 y T1021.001 como parte del perfil adversario. Sin embargo, estas técnicas no fueron reconocidas explícitamente en el análisis realizado por el sistema. En su lugar, se generaron descripciones asociadas a otras técnicas del marco MITRE ATT&CK, que no se encontraban entre las acciones efectivamente configuradas en la guía de emulación.

Este fenómeno puede deberse a múltiples causas:

- La naturaleza ambigua o incompleta de algunos logs, que no siempre contienen suficiente evidencia directa para inferir la técnica específica aplicada.
- Limitaciones del propio modelo LLM, que puede tender a seleccionar técnicas más genéricas o frecuentes en su entrenamiento, en detrimento de aquellas realmente ejecutadas pero menos documentadas.
- Ausencia de señales explícitas en el entorno simulado, donde la ejecución de una técnica puede no generar trazas fácilmente detectables por Wazuh o por los sensores configurados.
- Disociación semántica entre el lenguaje de los logs y las descripciones sacadas en el contexto, lo que complica el proceso de mapeo técnico en ausencia de reglas de correlación formales.

Este desfase no implica necesariamente un error del sistema, pero sí subraya la necesidad de complementar el análisis basado en LLMs con mecanismos adicionales de validación y correlación cruzada, especialmente si se desea utilizar este tipo de sistemas en entornos donde se requiere un alto nivel de fiabilidad operativa.

Además, se ha observado que algunos documentos recuperados (especialmente del catálogo CAPEC) carecían de contenido útil, lo que pone de manifiesto la necesidad de implementar mecanismos de filtrado o enriquecimiento adicional en el componente RAG, concretamente al crear la base de datos del contexto, para evitar la incorporación de entradas vacías o poco informativas.

En conjunto, este proyecto ha demostrado que los modelos de lenguaje generativo, cuando están correctamente integrados con sistemas de recuperación semántica y fuentes estructuradas de conocimiento, pueden aportar un valor real en el análisis y respuesta ante ciberincidentes, contribuyendo a la toma de decisiones de forma más ágil y fundamentada.

Asimismo, se ha abierto un espacio para reflexionar sobre la aplicabilidad práctica de los LLMs en contextos operativos, considerando aspectos como la confianza, la auditabilidad

y la interacción con analistas humanos. Más allá de su capacidad técnica, su adopción requerirá un equilibrio entre automatización y supervisión experta.

Por todo ello, el sistema desarrollado representa una aportación innovadora en el ámbito de la ciberdefensa, combinando tecnologías emergentes de IA generativa con marcos consolidados de seguridad operativa. Los resultados obtenidos sientan las bases para futuras mejoras, tanto en su rendimiento técnico como en su integración en entornos reales, y marcan una línea prometedora de evolución en el uso de IA y del enfoque RAG en seguridad informática.

6.2. Líneas futuras

A partir del trabajo realizado, se abren diversas líneas de investigación y desarrollo que podrían enriquecer y ampliar el alcance de este proyecto.

Una de las más relevantes consiste en evaluar el sistema en entornos reales de producción, donde la diversidad y volumen de los logs generados permitirían comprobar la robustez del modelo frente a datos menos estructurados, con ruido y múltiples orígenes. Esta validación permitiría también ajustar los mecanismos de inferencia y contextualización ante amenazas no simuladas, identificando nuevas necesidades operativas y casos de uso reales.

Otra mejora clave se refiere a la correlación precisa entre eventos observados y técnicas efectivamente ejecutadas. Como se ha identificado en ambos escenarios evaluados, el sistema (incluso con RAG) muestra dificultades para mapear los eventos registrados con las técnicas exactas de emulación, lo que limita la precisión del análisis. En este sentido, se propone desarrollar mecanismos de correlación reforzada basados en reglas, aprendizaje supervisado o fusión de múltiples fuentes (por ejemplo, información de telemetría, secuencias temporales o relaciones causales). Incorporar etiquetas semánticas o trazas de ejecución más ricas podría ayudar a reducir esta discrepancia.

Asimismo, se considera muy prometedora la posibilidad de ampliar y diversificar la base de conocimiento del sistema RAG, incorporando nuevas fuentes más orientadas al contexto real de organizaciones específicas. Esto incluiría informes de inteligencia de amenazas (Threat Intelligence Reports), reportes forenses, boletines de vulnerabilidades, políticas internas, o incluso registros históricos de incidentes documentados. Esta evolución permitiría que las respuestas generadas fueran aún más adaptadas, precisas y relevantes para distintos perfiles organizativos.

En paralelo, una línea natural de evolución técnica consiste en integrar el sistema con plataformas SOAR (Security Orchestration, Automation and Response). Esta integración permitiría automatizar parcialmente la ejecución de los COAs generados, bajo supervisión humana, cerrando así el ciclo de respuesta de forma ágil y orquestada. Esta arquitectura facilitaría una respuesta más rápida y coherente ante amenazas en tiempo real.

Del mismo modo, sería interesante explorar técnicas avanzadas de aprendizaje continuo o adaptación incremental, que permitan al modelo incorporar nuevas amenazas, patrones o cambios normativos sin necesidad de un reentrenamiento completo. Esto es especialmente importante en un dominio tan dinámico como la ciberseguridad, donde los modelos deben evolucionar junto con el ecosistema de amenazas.

Por último, se plantea como línea futura la creación de una interfaz interactiva y transparente, orientada al analista humano. Esta interfaz permitiría visualizar las decisiones

del modelo, modificar criterios de análisis, personalizar reglas de recuperación, o incluso contribuir al aprendizaje del sistema mediante correcciones y sugerencias. Este enfoque favorecería una colaboración fluida entre inteligencia artificial y expertos humanos, potenciando la confianza y explicabilidad del sistema.

En resumwn, estas líneas de trabajo no solo amplían las posibilidades técnicas del sistema propuesto, sino que también apuntan a una evolución del paradigma actual de ciberdefensa, en el que las soluciones automatizadas e inteligentes no sustituyen, sino que complementan, potencian y aceleran la labor de los profesionales humanos. La sinergia entre ambos elementos será, sin ninguna duda, uno de los pilares fundamentales para afrontar los retos de seguridad del futuro digital.

Bibliografía

[1] Wazuh official website. https://wazuh.com/. Accedido en 2025.

[2] MITRE Corporation. Mitre att&ck framework. https://attack.mitre.org/, 2024. Accedido en 2025.

[3] Mitre. Mitre caldera. https://caldera.mitre.org/. Accedido en 2025.

[4] Universidad Politécnica de Madrid. Ctf25 - upmdrive. https://drive.upm.es/s/g356YKZY7jXMYrv, 2024. Accedido en 2025.

[5] Tom Brown et al. Language models are few-shot learners. *NeurIPS*, 2020.

[6] Barret Zoph et al. Emergent abilities of large language models. *TMLR*, 2022.

[7] Shervin Minaee et al. Large language models: A survey. 2024.

[8] Gabriel de Jesus Coelho da Silva and Carlos Becker Westphall. A survey of large language models in cybersecurity. 2024.

[9] Farzad Nourmohammadzadeh Motlagh et al. Llms in cybersecurity: State-of-the-art. 2024.

[10] Yagmur Yigit et al. Review of generative ai methods in cybersecurity. 2024.

[11] A. Shestov et al. Finetuning llms for vulnerability detection. 2024.

[12] Z. Li et al. Llm-assisted static analysis for detecting vulnerabilities. 2024.

[13] M. Sultana et al. Evaluation of llms for cyber operation automation. In *IEEE CNS*, 2023.

[14] V. Rawte et al. A survey of hallucination in large foundation models. 2023.

[15] S Selva Kumar, Afifah Khan Mohammed Ajmal Khan, Imadh Ajaz Banday, Manikantha Gada, and Vibha Venkatesh Shanbhag. Overcoming llm challenges using rag-driven precision in coffee leaf disease remediation. In *2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS)*, 2024.

[16] J. Qi et al. Loggpt: Exploring chatgpt for log-based anomaly detection. 2023.

[17] P. Manakul et al. Self-checkgpt: Zero-resource black-box hallucination detection. 2023.

[18] OpenAI. Gpt-4 turbo and gpt-4 model. https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4, 2024. Accedido en 2025.

[19] G. Team et al. Gemma: Open models based on gemini research. 2024.

[20] Hugo Touvron et al. Llama: Open and efficient foundation language models. 2023.

[21] Albert Q Jiang et al. Mixtral of experts. 2024.

[22] E. Strubell et al. Energy and policy considerations for deep learning in nlp. In *ACL*, 2019.

[23] E. M. Bender et al. On the dangers of stochastic parrots. In *ACM FAccT*, 2021.

[24] A. Wang et al. Efficient training of llms: A survey. 2020.

[25] Amrita Bhattacharjee et al. Towards llm-guided causal explainability. 2024.

[26] V. Juttner et al. Chatids: Explainable cybersecurity using generative ai. 2023.

[27] Julian Ambacher. Designing a user-friendly and optimized version of a user interface for an llm, 2024.

[28] Sabrina Aquino. What is rag: Understanding retrieval-augmented generation. https://qdrant.tech/articles/what-is-rag-in-ai/, 2024.

[29] Bert model documentation. https://huggingface.co/docs/transformers/model_doc/bert. Accedido en 2025.

[30] New embedding models and api updates. https://openai.com/index/new-embedding-models-and-api-updates/. Accedido en 2025.

[31] Jeff Johnson, Matthijs Douze, and Hervé Jégou. The faiss library. https://faiss.ai/, 2019.

[32] Nikita Kathare, O. Vinati Reddy, and Vishalakshi Prabhu. A comprehensive study of elastic search. *Journal of Research in Science and Engineering (JRSE)*, 2022.

[33] Chroma. Chromadb - open-source embedding database. https://www.trychroma.com/, 2024. Accedido en 2025.

[34] Knn vs ann - azure cosmos db. https://learn.microsoft.com/es-es/azure/cosmos-db/gen-ai/knn-vs-ann. Accedido en 2025.

[35] Sayantan Pal, Maiga Chang, and Maria Fernandez Iriarte. Summary generation using natural language processing techniques and cosine similarity. In *International Conference on Intelligent Systems Design and Applications*. Springer, 2021.

[36] Annoy integration with langchain. https://python.langchain.com/docs/integrations/vectorstores/annoy/. Accedido en 2025.

[37] Langgraph by langchain. https://www.langchain.com/langgraph. Accedido en 2025.

[38] Cisco. Splunk. https://www.splunk.com/. Accedido en 2025.

[39] Elastic. Elastic security. https://www.elastic.co/es/security. Accedido en 2025.

[40] Wazuh. Wazuh agent groups and centralized configuration. https://wazuh.com/blog/agent-groups-and-centralized-configuration/. Accedido en 2025.

[41] Wazuh documentation. https://documentation.wazuh.com/current/. Accedido en 2025.

[42] Wazuh compliance documentation. https://documentation.wazuh.com/current/compliance/index.html. Accedido en 2025.

[43] Muhammad Mudassar Yamin et al. Applications of llms for generating cyber security exercise scenarios. 2024.

[44] Dipkamal Bhusal et al. Secure: Benchmarking large language models for cybersecurity. 2024.

[45] Marco Simoni et al. Morse: Bridging the gap in cybersecurity expertise with rag. 2021.

[46] Jonathan Pan et al. Raglog: Log anomaly detection using retrieval augmented generation. 2024.

[47] Jiarui Li et al. Enhancing soc efficiency with multi-model integration. 2024.

[48] Mikko Lempinen et al. Chatbot for assessing system security with openai gpt-3.5. 2023.

[49] Mehrdad Kaheh et al. Cyber sentinel: Gpt-4 for security tasks. 2023.

[50] Jie Zhang et al. When llms meet cybersecurity: A systematic literature review. 2024.

[51] Mitre. Cve. https://www.cve.org/. Accedido en 2025.

[52] Ollama. Llama3.1:8b. https://ollama.com/library/llama3.1:8b, 2024. Accedido en 2025.

[53] Python Software Foundation. Python programming language. https://www.python.org/, 2024. Accedido en 2025.

[54] LangChain Inc. Langchain - framework for developing llm-powered applications. https://www.langchain.com/, 2024. Accedido en 2025.

[55] LangChain Inc. Langsmith - debug, evaluate, and monitor llm applications. https://www.langchain.com/langsmith, 2024. Accedido en 2025.

[56] Docker Inc. Docker - empowering app development for developers. https://www.docker.com/, 2024. Accedido en 2025.

[57] Ollama. Ollama - run llms locally. https://ollama.com/, 2024. Accedido en 2025.

[58] Joseph A Izzo, Namhee Kim, Shereef Elmetwaly, and Tamar Schlick. Rag: an update to the rna-as-graphs resource. *BMC bioinformatics*, 2011.

[59] J. et al. Devlin. Bert: Pre-training of deep bidirectional transformers. 2019.

[60] J. et al. Johnson. Billion-scale similarity search with gpus. faiss. 2019.

[61] Elasticsearch documentation. opensearch and semantic search in large databases, 2023.

[62] European Union Agency for Cybersecurity (ENISA). Threat landscape 2023. https://www.enisa.europa.eu/topics/cyber-threats/threat-landscape, 2023. Accedido en 2025.

[63] Center for Internet Security (CIS). Cis benchmarks. https://www.cisecurity.org/cis-benchmarks, 2024. Accedido en 2025.

[64] Mandiant. Mandiant threat intelligence reports. https://www.mandiant.com/resources, 2024. Accedido en 2025.

[65] Palo Alto Networks – Unit 42. Unit 42 threat intelligence reports. https://unit42.paloaltonetworks.com/category/threat-research/, 2024. Accedido en 2025.

[66] CrowdStrike. Global threat report 2024. https://www.crowdstrike.com/en-us/global-threat-report/, 2025. Accedido en 2025.

[67] National Institute of Standards and Technology (NIST). National vulnerability database (nvd). https://nvd.nist.gov/, 2024. Accedido en 2025.

[68] U.S. Department of Commerce. Nist. https://www.nist.gov/. Accedido en 2025.

[69] U.S. Department of Commerce. Security and privacy controls for information systems and organizations. https://doi.org/10.6028/NIST.SP.800-53r5, 2020.

[70] U.S. Department of Commerce. Incident response recommendations and considerations for cybersecurity risk management. https://doi.org/10.6028/NIST.SP.800-61r3, 2025.

[71] Murugiah Souppaya Karen Kent. Guide to computer security log management. https://doi.org/10.6028/NIST.SP.800-92, 2006.

[72] Mitre. D3fend. https://d3fend.mitre.org/. Accedido en 2025.

[73] Mitre. Capec common attack pattern enumeration and classification. https://capec.mitre.org/. Accedido en 2025.

[74] Mitre. Cwe common weakness enumartion. https://cwe.mitre.org/. Accedido en 2025.

[75] Tolga Şakar and Hakan Emekci. Maximizing rag efficiency: A comparative analysis of rag methods. *Natural Language Processing*, 2025.

[76] Wazuh. Adversary emulation with caldera and wazuh. https://wazuh.com/blog/adversary-emulation-with-caldera-and-wazuh/, 2024. Accedido en 2025.

[77] Wazuh. Migrating from ossec to wazuh. https://wazuh.com/blog/migrating-from-ossec-to-wazuh/, 2023. Accedido en 2025.

[78] Antonio Matas. Diseño del formato de escalas tipo likert: un estado de la cuestión. *Revista electrónica de investigación educativa*, 2018.

[79] Andrea Valenzuela. Cuantificación para grandes modelos lingüísticos (llm): Reduce eficazmente el tamaño de los modelos de ia, 2024.

[80] Rui Ye et al. Openfedllm: Training llms on decentralized private data. 2024.

Anexo A: Aspectos éticos, económicos, sociales y ambientales

A1. Introducción

Este Trabajo de Fin de Grado propone el diseño e implementación de un sistema que emplea modelos de lenguaje avanzados (LLMs), apoyados en un enfoque de Generación Aumentada por Recuperación (RAG), para analizar logs generados por un SIEM y generar recomendaciones automatizadas frente a ciberamenazas. Este sistema pretende reducir la carga cognitiva de los analistas, acelerar la detección de amenazas y proponer Cursos de Acción (COAs) pertinentes. Esta solución aborda no solo un problema técnico, sino también una necesidad creciente en el ámbito de la ciberseguridad, con importantes implicaciones éticas, sociales, económicas y medioambientales. La automatización y mejora en la toma de decisiones que ofrece este proyecto exige una reflexión crítica sobre su impacto en los profesionales, en la sociedad y en el uso responsable de las tecnologías.

A2. Descripción de impactos relevantes relacionados con el proyecto

Durante el desarrollo del proyecto, se han identificado varios impactos relevantes:

- Ético: La integración de modelos LLM implica la responsabilidad de asegurar la transparencia y verificabilidad de las decisiones automáticas. En contextos sensibles como la ciberseguridad, los riesgos asociados a errores, sesgos o "alucinaciones" del modelo deben minimizarse. Por ello, se ha priorizado el uso de modelos open-source (Llama 3.1), que permiten auditorías y ajustes, garantizando mayor control sobre su funcionamiento.
- Social: El sistema puede contribuir a democratizar el acceso a herramientas de defensa avanzada, beneficiando especialmente a pequeñas organizaciones que no cuentan con equipos de ciberseguridad especializados. Sin embargo, también debe considerarse su potencial impacto sobre el empleo, al automatizar tareas tradicionalmente realizadas por analistas.
- Económico: La adopción de este tipo de sistemas puede reducir costes operativos en los SOC (Security Operations Centers) al minimizar el tiempo de análisis y respuesta. Asimismo, el uso de software libre (Wazuh, Llama 3.1, ChromaDB) favorece la sostenibilidad económica del proyecto y su escalabilidad sin incurrir en licencias propietarias.
- Ambiental: Si bien el uso de LLMs implica un consumo significativo de recursos

computacionales [24], el enfoque local y optimizado (ejecución mediante Ollama y uso de embeddings preprocesados) reduce la dependencia de servicios en la nube y disminuye el impacto energético derivado de transferencias y procesos remotos.

Los principales grupos de interés identificados son los analistas de ciberseguridad, responsables de TI, pequeñas y medianas empresas, desarrolladores de herramientas de defensa, y organismos que definen normativas en ciberseguridad.

A3. Análisis detallado de alguno de los impactos

El uso de modelos de lenguaje a gran escala conlleva un importante consumo de recursos computacionales, tanto durante su entrenamiento como en su despliegue en sistemas operativos reales. Este aspecto no solo representa un reto económico —especialmente para organizaciones con recursos limitados—, sino también un impacto ambiental, debido a la huella energética asociada al uso intensivo de infraestructuras de computación. La sostenibilidad, por tanto, emerge como una dimensión crítica en el diseño e implementación de soluciones basadas en IA.

Ante esta situación, se han incorporado al diseño del sistema estrategias orientadas a mitigar el impacto energético sin comprometer la calidad de los resultados. En primer lugar, se ha optado por utilizar modelos open-source como Llama 3.1, que ofrecen un equilibrio notable entre precisión y eficiencia operativa [20]. A diferencia de modelos propietarios más pesados, estos permiten su ejecución en entornos locales con hardware optimizado, evitando así el uso intensivo de grandes centros de datos.

Asimismo, se considera el uso de técnicas de compresión de modelos, como la cuantificación [79] o el *pruning*, que reducen el tamaño y el consumo computacional de los modelos sin degradar significativamente su rendimiento. Estas técnicas no solo permiten una inferencia más eficiente, sino que abren la puerta a la ejecución de LLMs en entornos con recursos restringidos, como dispositivos IoT o sistemas embebidos, aumentando la escalabilidad y sostenibilidad de las soluciones.

Otra línea prometedora en términos de sostenibilidad y privacidad es el entrenamiento federado [80]. Aunque aún no está ampliamente aplicado en el campo de la ciberseguridad, este paradigma permite distribuir el proceso de entrenamiento entre múltiples nodos o clientes, reduciendo la necesidad de transferir datos sensibles a servidores centrales. Este enfoque no solo disminuye el consumo energético en centros de datos, sino que también refuerza el cumplimiento normativo en cuanto a la protección de datos, abordando simultáneamente preocupaciones éticas y legales.

En conjunto, estas estrategias demuestran que es posible desarrollar soluciones basadas en LLMs que respeten principios de sostenibilidad y responsabilidad ambiental, sin renunciar a un alto rendimiento. El sistema propuesto en este TFG ha sido concebido con estos principios en mente, priorizando modelos y arquitecturas eficientes que pueden adaptarse a distintas escalas de implementación.

A4. Conclusiones

Desde una perspectiva ética, económica, social y ambiental, el proyecto representa un avance equilibrado en la aplicación de la inteligencia artificial en ciberseguridad. Se ha procurado que las decisiones del sistema sean auditables y justificadas, favoreciendo un uso

responsable de las tecnologías. La elección de herramientas de código abierto ha reducido tanto el impacto económico como ambiental, permitiendo su adopción por organizaciones de distintos tamaños. En el plano social, el sistema tiene potencial para democratizar el acceso a defensas avanzadas, siempre que se acompañe de formación adecuada para su correcta interpretación y uso.

La consideración de criterios de sostenibilidad ha aportado un valor añadido tangible al proyecto, orientando sus decisiones técnicas hacia soluciones más accesibles, responsables y replicables. Esta aproximación no solo fortalece la validez técnica del trabajo, sino también su aplicabilidad real en entornos profesionales.

Anexo B: Presupuesto económico

En el contexto de este trabajo, se ha llevado a cabo una labor intensiva de análisis, diseño, implementación, validación y documentación de un sistema híbrido de ciberdefensa. Este sistema integra modelos de lenguaje avanzados con un enfoque RAG para el análisis de logs generados por un SIEM, proponiendo recomendaciones de respuesta a incidentes de seguridad.

El desarrollo ha implicado un uso significativo de recursos computacionales, herramientas de software especializado y material de apoyo técnico, además de una considerable dedicación personal por parte del autor. A continuación, se presenta una estimación de los costes asociados al proyecto, con el fin de valorar su viabilidad económica y su alineación con los criterios de sostenibilidad establecidos en las acreditaciones EUR-ACE y ABET.

COSTE DE MANO DE OBRA (coste directo)	**Horas**	**Precio/hora**	**Total**
	335	13 €	**4.355 €**

COSTE DE RECURSOS MATERIALES (coste directo)	**Precio de compra**	**Uso en meses**	**Amortización (en años)**	**Total**
Portátil personal (procesador avanzado, GPU local para LLMs, software incluido)	1.800,00 €	6	5	180,00 €
Impresora y escáner multifunción (para documentación)	120,00 €	6	5	12,00 €

COSTE TOTAL DE RECURSOS MATERIALES	**200,00 €**

GASTOS GENERALES (costes indirectos)	15 %	sobre CD	**705,00 €**
BENEFICIO INDUSTRIAL	6 %	sobre CD + CI	**324,30 €**

MATERIAL FUNGIBLE	
Impresión	**100,00 €**
Encuadernación	**300,00 €**

SUBTOTAL PRESUPUESTO		**5.984,30 €**
IVA APLICABLE	21 %	**1.256,70 €**

TOTAL PRESUPUESTO	**7.241,00 €**

Tabla 6.1: Presupuesto económico