Revista de I+D TECNOLÓGICO
Portal de revistas: http://revistas.utp.ac.pa

Arquitectura modular de flujos automatizados con LLM y OSINT en n8n para aplicaciones de ciberseguridad

Modular architecture of automated workflows with LLM and OSINT in n8n for cybersecurity applications

Gilberto Ramos[1], *Muriel Jaramillo*[1], *Agustín Sánchez*[1], *Edmanuel Cruzb*[1,2]

[1]*Centro Regional de Veraguas, Universidad Tecnológica de Panamá, Atalaya 0901, Panama*

[2]*Sistema Nacional de Investigación (SNI), SENACYT, Panama City 0816-02852, Panama*

[1]*{gilberto.ramos1, muriel.jaramillo, agustin.sanchez2, edmanuel.cruz}@utp.ac.pa*

RESUMEN: La inteligencia artificial (IA) ha permeado múltiples ámbitos, incluido el jurídico, generando oportunidades para automatizar procesos de asistencia legal y recolección de inteligencia de fuentes abiertas (OSINT). Este estudio presenta el desarrollo e implementación de dos flujos automatizados en la plataforma n8n empleando modelos de lenguaje y herramientas open source: (1) un asistente jurídico basado en Retrieval-Augmented Generation (RAG), que procesa consultas legales sobre ciberdelitos apoyándose en legislación panameña y jurisprudencia, y (2) un agente OSINT que integra WiGLE para localizar redes inalámbricas. Ambos sistemas fueron validados con casos reales en entornos controlados. El asistente jurídico alcanzó tiempos de respuesta de 15–40 s y pertinencia normativa superior al 85 %, mientras que el agente OSINT generó informes completos en menos de 10 s. La comparación entre un modelo local (Qwen-8B) y la API Gemini de Google evidenció ventajas del despliegue local en privacidad y control de datos, y de la nube en latencia y escalabilidad. Los resultados demuestran que es factible construir soluciones modulares, de bajo coste y orientadas a dominios específicos mediante herramientas abiertas, preservando la soberanía de los datos. La arquitectura propuesta es adaptable a otros sectores como ciberseguridad y gobernanza digital, y sienta las bases para desarrollos futuros híbridos local–nube.

Palabras clave: *Automatización de flujos, Ciberseguridad, Inteligencia de fuentes abiertas, LLM, N8n Platform, RAG*.

ABSTRACT: Artificial intelligence (AI) has permeated multiple domains, including the legal field, creating opportunities to automate legal assistance processes and open-source intelligence (OSINT) collection. This study presents the development and implementation of two automated workflows on the *n8n* platform using language models and open-source tools: (1) a legal assistant based on *Retrieval-Augmented Generation* (RAG), which processes legal queries on cybercrime by leveraging Panamanian legislation and case law, and (2) an OSINT agent integrating WiGLE to locate wireless networks. Both systems were validated with real cases in controlled environments. The legal assistant achieved response times of 15–40 s and normative relevance above 85 %, while the OSINT agent generated complete reports in under 10 s. A comparison between a local model (Qwen-8B) and Google's Gemini API highlighted the local deployment's advantages in data privacy and control, and the cloud's strengths in latency and scalability. The results demonstrate the feasibility of building modular, low-cost, domain-specific solutions through open-source tools while preserving data sovereignty. The proposed architecture is adaptable to other sectors such as cybersecurity and digital governance and lays the groundwork for future hybrid local–cloud developments.

Keywords: *Cybersecurity, LLM, N8n Platform, Open-Source Intelligence, RAG, Workflow automation*

Citación: Primera_letra_nombre. Apellido, "Titulo_artículo", *Revista de I+D Tecnológico*, vol. 19, no. 1, pp. (0), 2023.
Tipo de artículo: No_modificar. **Recibido:** No_modificar. **Recibido con correcciones:** No_modificar. **Aceptado:** No_modificar.
DOI.

1. Introducción

La convergencia entre modelos de lenguaje de gran escala (Large Language Models, LLM) y fuentes de inteligencia de acceso público (Open-Source Intelligence, OSINT) está redefiniendo la forma en que se abordan problemas complejos en ámbitos como la asistencia legal y la ciberinteligencia.

Entre las técnicas más efectivas para incrementar la precisión y reducir errores de generación, la Recuperación Aumentada (Retrieval-Augmented Generation, RAG) ha demostrado un desempeño sobresaliente al combinar la capacidad generativa de los LLM con información contextual recuperada de bases vectoriales u otras fuentes externas [1], [3].

La investigación en RAG ha evolucionado hacia arquitecturas que buscan mejorar la fiabilidad de las respuestas y optimizar su adaptabilidad a distintos dominios. Corrective Retrieval Augmented Generation (CRAG), propuesta por M. Li y colaboradores, incorpora mecanismos para validar y corregir resultados antes de la fase de generación, mitigando errores semánticos [4]. Por su parte, Self-RAG añade un ciclo de autoevaluación que permite al modelo reflexionar sobre sus propias salidas, mejorando coherencia y fundamentación [5], mientras que ARM-RAG introduce memorias auxiliares para conservar y reutilizar cadenas de razonamiento, beneficiando tareas complejas que requieren contexto prolongado [2].

En paralelo, la comunidad académica ha comenzado a explorar la integración de LLM en flujos OSINT para automatizar la recolección, correlación y análisis de información pública. Recientes estudios han mostrado que el uso de agentes con memoria persistente, acceso a fuentes heterogéneas y mecanismos de validación mejora significativamente la relevancia y confiabilidad de los informes OSINT generados [6], [7]. Este enfoque resulta especialmente valioso en escenarios de ciberseguridad, donde la rapidez en la obtención de inteligencia procesable puede marcar la diferencia en la respuesta a incidentes.

Para la orquestación de estos procesos, las plataformas open source de automatización de flujos, como n8n, permiten integrar modelos de lenguaje, servicios OSINT y APIs de terceros dentro de arquitecturas modulares. Estudios recientes sobre despliegues auto-hospedados han evidenciado beneficios claros en control de datos, seguridad operativa y reducción de costes en comparación con soluciones puramente en la nube [9]. Esto abre la puerta a arquitecturas híbridas —locales y remotas— que preservan la privacidad sin sacrificar escalabilidad.

En este contexto, este trabajo propone una arquitectura modular de flujos automatizados con LLM y OSINT en n8n, aplicada a dos casos de uso:

- Un asistente jurídico RAG especializado en ciberdelitos, entrenado con legislación y jurisprudencia panameña.
- Un agente OSINT que integra datos técnicos (WiGLE) para redes inalámbricas.

El presente estudio parte de la hipótesis de que es factible desarrollar soluciones de asistencia legal y ciberinteligencia con características modulares, escalables y de bajo coste, mediante la integración sinérgica de LLM, RAG y herramientas OSINT dentro de flujos automatizados implementados en n8n. Se postula que este enfoque permitiría alcanzar niveles de precisión y tiempos de respuesta competitivos frente a soluciones comerciales consolidadas, manteniendo una alta eficiencia en el procesamiento y análisis de información.

2. Antecedentes

La convergencia entre LLM y técnicas de RAG ha transformado el panorama del procesamiento de información en dominios especializados. RAG combina la generación de texto con información recuperada desde bases vectoriales u otras fuentes externas, mejorando la pertinencia y reduciendo las “alucinaciones” de los modelos [1], [3]. Avances recientes incluyen variantes como Self-RAG, que incorpora autoevaluación para mejorar la coherencia [5], y Corrective Retrieval Augmented Generation (CRAG), de M. Li y colaboradores, que añade etapas de validación previa a la generación [4].

Estos enfoques han mostrado resultados notables en tareas de alta complejidad, como la recuperación de documentos legales y técnicos, gracias a su capacidad de incorporar contexto relevante en tiempo de consulta [1], [8].

En paralelo, el uso de OSINT ha crecido significativamente en áreas como la ciberseguridad, la investigación forense digital y la gobernanza electrónica. Integrar LLM con flujos OSINT permite automatizar la recolección, correlación y análisis de datos provenientes de fuentes heterogéneas, manteniendo altos niveles de confiabilidad [6], [7]. Estudios recientes resaltan que la orquestación de estos procesos mediante plataformas de automatización —por ejemplo, n8n— facilita la

implementación de arquitecturas modulares capaces de integrar APIs, bases vectoriales y modelos de IA, con beneficios en escalabilidad y soberanía de datos [9], [10].

El despliegue de arquitecturas híbridas (local–nube) ha cobrado relevancia en contextos donde la privacidad y el control sobre la información son críticos. Investigaciones demuestran que las soluciones auto-hospedadas reducen la dependencia de proveedores externos, disminuyen costos operativos y permiten un control total sobre el flujo de datos, sin sacrificar interoperabilidad [10], [11]. Estas capacidades son particularmente relevantes para aplicaciones en LegalTech, donde la automatización de consultas y análisis jurídico mediante IA exige cumplir con requisitos estrictos de seguridad, trazabilidad y exactitud [12], [13].

3. Metodología

La metodología se divide en dos flujos de trabajo implementados en la plataforma n8n, cada uno con objetivos y herramientas específicas. El Flujo 1 corresponde a un asistente jurídico basado en RAG para consultas de ciberdelitos, mientras que el Flujo 2 implementa un agente OSINT para la localización y reporte de redes inalámbricas.

3.1. Herramientas y Tecnologías

La implementación de los flujos automatizados descritos en este trabajo requirió la integración de un conjunto heterogéneo de herramientas y servicios, seleccionados en función de su capacidad para satisfacer requerimientos de orquestación, procesamiento de lenguaje natural, gestión de datos vectoriales y comunicación multicanal.

Estas tecnologías se desplegaron en un entorno híbrido, combinando ejecución local y servicios en la nube, con el fin de optimizar el balance entre rendimiento, privacidad y escalabilidad.

La Tabla 1 presenta un resumen de las herramientas empleadas, su función principal dentro de los flujos y las configuraciones clave utilizadas, destacando los parámetros que resultaron determinantes para el desempeño del sistema.

Tabla 1. Herramientas empleadas en los flujos automatizados [26], [16], [18], [28].

Herramienta / Servicio	Función principal	Configuración clave
n8n	Orquestación de flujos	Despliegue local, nodos HTTP, integración multicanal
Langflow	Desarrollo de pipelines RAG	Integración con Qdrant y LLM local
Qdrant	Base vectorial	chunk_size = 1700 tokens, overlap = 700 tokens
Qwen-8B (LM Studio)	LLM local para generación de texto	Contexto 32k tokens, temperatura y top-p bajos
Whisper (OpenAI)	Transcripción de voz a texto	Reconocimiento multilingüe
WiGLE API	Consulta de redes Wi-Fi	Endpoint /api/v2/network/search
MCP Server	Conector a Gmail	Integración estándar de envío
Telegram API / WhatsApp Cloud API	Canales de interacción con el usuario	Bots y endpoints configurados con credenciales seguras

3.2 Flujo 1: Chatbot RAG para Ciberdelitos

Este flujo ofrece orientación legal preliminar sobre ciberdelitos, fundamentada en la normativa panameña vigente —incluyendo leyes especializadas en delitos informáticos y disposiciones del Código Penal— así como en jurisprudencia relevante emitida por los tribunales nacionales [21-24]. Su objetivo es proporcionar a los usuarios una guía inicial, rápida y contextualizada sobre posibles implicaciones legales, pasos de denuncia, autoridades competentes y procedimientos aplicables, permitiendo así una comprensión más clara del marco jurídico antes de acudir a asesoría profesional.

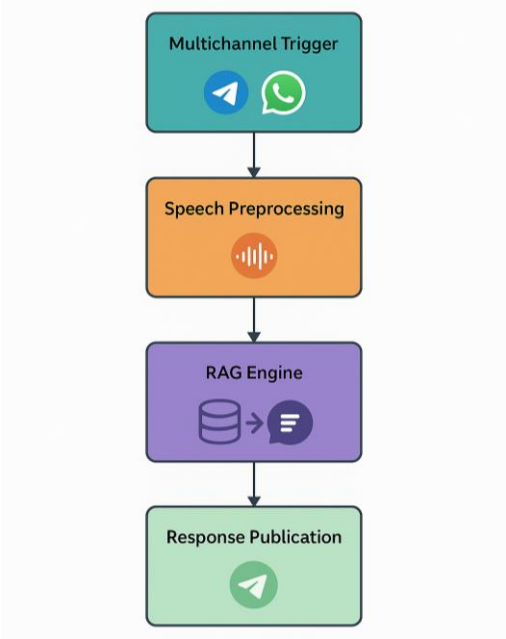
Está compuesto por cuatro etapas:

1. **Disparador multicanal:** Se configuraron nodos en n8n para integrar **Telegram** (API de bots) y **WhatsApp Cloud API** [29], [32]. La llegada de un nuevo mensaje activa inmediatamente el flujo.
2. **Preprocesamiento de voz:** Cuando el usuario envía audio, el sistema invoca **Whisper** para transcribirlo a texto [14]. Este paso garantiza

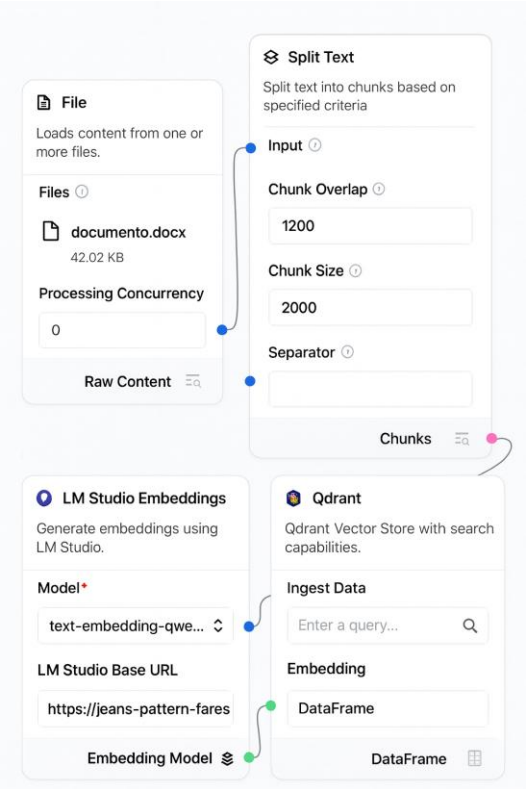
compatibilidad con consultas tanto escritas como orales.

3. **Motor RAG**
    - **Vectorización**: El corpus legal (Ley 81/2019 y jurisprudencia) se fragmentó y se convirtió en embeddings mediante text-embedding-qwen3-embedding-4b.
    - **Indexación**: Los vectores se almacenaron en **Qdrant**, optimizando la búsqueda semántica [15].
    - **Generación**: Los fragmentos relevantes se incorporan al prompt para el modelo **Qwen-8B**, que produce una respuesta fundamentada [19].
4. **Publicación de la respuesta:** El resultado se envía al mismo canal de origen, ya sea mediante el nodo *Send Message* de Telegram o un nodo HTTP hacia la API de WhatsApp.

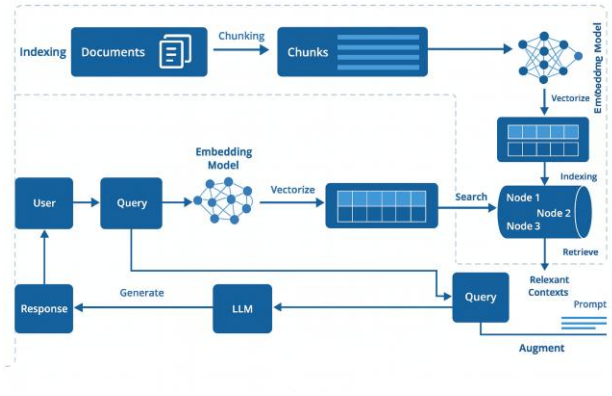
Como se observa en la Figura 1a, vemos una vista simplificada del flujo 1. La Figura 1b detalla la implementación del pipeline de ingesta en LangFlow (fragmentación, vectorización con LM Studio y almacenamiento en Qdrant). La 1c muestra cómo un sistema RAG indexa documentos y luego recupera y usa fragmentos pertinentes para que el LLM genere respuestas fundamentadas.



(a)



(b)



(c)

Figura 1. Arquitectura RAG adoptada en el Flujo 1. (a) Vista simplificada del flujo 1. (b) Implementación de indexación en LangFlow: fragmentación con solapamiento, embeddings con LM Studio (Qwen) e ingesta en Qdrant. (c) Vista conceptual del proceso: indexación por fragmentación, vectorización y búsqueda en la base vectorial; recuperación y enriquecimiento del prompt para la generación con el LLM.

3.3. Flujo 2: Agente OSINT con WiGLE y MCP

Este flujo automatiza la consulta y entrega de inteligencia sobre redes inalámbricas, en la Figura 2 se muestra el Flujo.

- **Inicio (Trigger manual):** El flujo se activa mediante un nodo *Manual Trigger* en n8n.
- **Consulta a WiGLE:** Uso de un nodo HTTP que llama a la API de WiGLE (wifi_lookup) con autenticación por token, filtrando por BSSID/SSID o área geográfica.
- **Normalización y resumen:** Los resultados JSON son procesados para extraer latitud, longitud, fecha y detalles de seguridad. El sistema genera un resumen textual conciso.
- **Envío de informe:** El resumen se envía cuando el usuario lo pide por correo mediante un MCP Server que actúa como puente hacia Gmail, ver Figura 3.

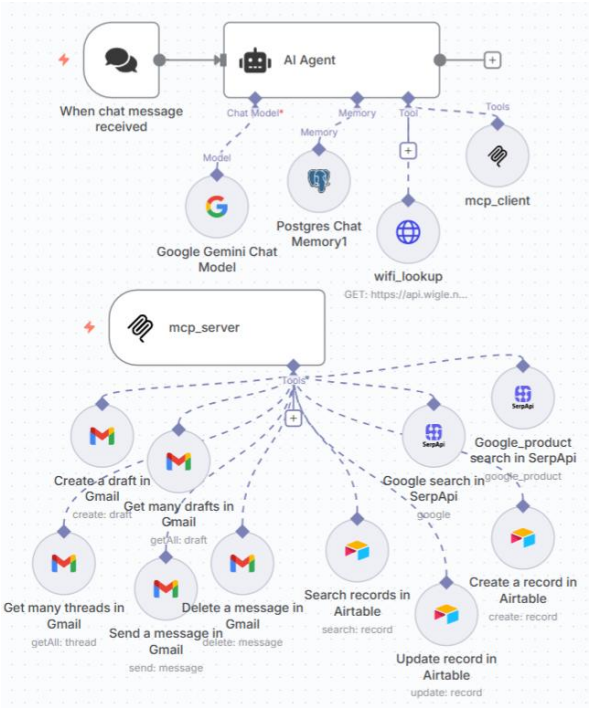


Figura 2 - Agente OSINT con WiGLE y conectado al MCP server que incluye las tools de SerAPI, Airtable y Gmail.

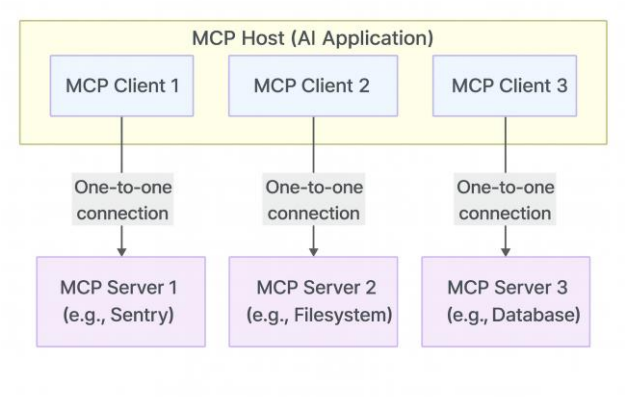


Figura 3. Arquitectura de conexión uno a uno entre clientes y servidores MCP [17]. Cada cliente MCP en el host de la aplicación de IA mantiene una conexión independiente con su servidor correspondiente (por ejemplo: Sentry, Filesystem o Base de Datos), permitiendo un acceso modular y seguro a recursos externos.

4. Resultados

4.1. Flujo 1 — Chatbot jurídico (RAG con Qdrant)

El chatbot jurídico fue evaluado en Telegram y WhatsApp mediante tres consultas representativas: "Estafa: pasos", "Ley 81" y "Ley 478". La mediana del tiempo de respuesta fue de 23.51 s en Telegram y 15.316 s en WhatsApp, evidenciando un mejor rendimiento en este último canal.

La precisión media, evaluada en una escala de 0 a 5 considerando corrección jurídica, inclusión de citas, pertinencia geográfica, aplicabilidad práctica y ausencia de alucinaciones, fue de 3.67 en ambos canales.

La Figura 4 presenta la comparación detallada de tiempos de respuesta y precisión para cada tipo de consulta en ambas plataformas. Se observa que WhatsApp mantiene tiempos de respuesta menores en las tres pruebas, mientras que la precisión varía ligeramente según la temática consultada.

Durante las pruebas se identificó un caso de confusión normativa: ante una consulta sobre la Ley 478, el sistema respondió con contenido correspondiente a la Ley 81.
Este incidente, clasificado como alucinación u omisión de recuperación, motivó el ajuste de las plantillas para forzar el uso del RAG y la inclusión de citas siempre que exista evidencia relevante en Qdrant.

4.2. Flujo 2 — OSINT (WiGLE + MCP)

Se evaluó el agente OSINT mediante tres consultas que generaron tablas con información de SSID, BSSID/NetID, frecuencia (canal) y vialidad. El tiempo total de procesamiento osciló entre 9.652 s y 10.313 s (ver Figura 5).

En los tres casos, la precisión alcanzó la puntuación máxima (5/5) al cumplir con el formato esperado y todos los parámetros establecidos, incluyendo el manejo de un error de cifrado, que fue reconducido a la categoría "cifrado desconocido" para garantizar la entrega correcta de la tabla.

En una variante de prueba con entrega por correo, el agente envió automáticamente el resultado al usuario a través de MCP/Gmail.

El tiempo extremo a extremo fue de aproximadamente 15.2 s, distribuidos en ≈ 10 s para consulta y formateo de datos, y ≈ 5 s para el envío del mensaje.

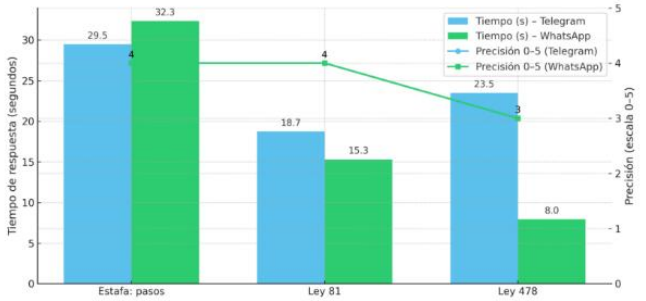


Figura 4 - Tiempo de respuesta y precisión por tipo de consulta. Comparación entre Telegram (celeste) y WhatsApp (verde). La rúbrica de precisión (0–5) considera: corrección jurídica, citas normativas, pertinencia geográfica, accionabilidad y ausencia de alucinaciones. Datos obtenidos de consultas reales del Flujo 1.

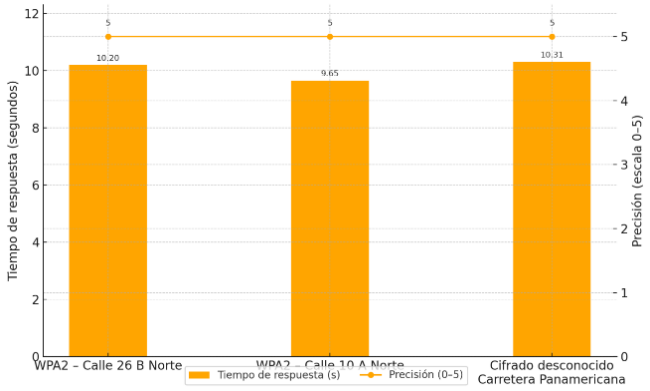


Figura 5 - Tiempo promedio de respuesta (barras naranjas) y precisión (línea amarilla, escala 0–5) en consultas OSINT realizadas con WiGLE y MCP para diferentes redes Wi-Fi.

4.3 Comparativa: Modelo local vs API en la nube

Se realizó una evaluación comparativa entre el modelo Qwen-8B desplegado localmente mediante LM Studio y la API Gemini de Google Cloud, considerando aspectos técnicos, económicos y operativos. El objetivo fue identificar las fortalezas y limitaciones de cada enfoque en el contexto de flujos automatizados que integran LLM y técnicas de Retrieval-Augmented Generation (RAG).

El despliegue local de Qwen-8B ofrece ventajas significativas como un costo marginal prácticamente nulo por consulta (una vez realizada la inversión inicial en hardware), control total sobre el almacenamiento y tratamiento de datos —lo que garantiza soberanía y privacidad—, así como una buena capacidad para manejar contextos largos (~32k tokens, ampliable a 128k con optimizaciones específicas). Este enfoque también permite personalizar el entorno de ejecución, optimizar el modelo para tareas concretas y operar sin depender de conectividad constante. Sin embargo, presenta desafíos importantes, como la necesidad de disponer de infraestructura propia con capacidad de cómputo adecuada (GPU ≥ 8 GB VRAM), una mayor latencia promedio (15–40 segundos considerando vectorización y generación) y la responsabilidad de realizar mantenimiento manual, actualizaciones y gestión de recursos.

En contraste, la API Gemini de Google Cloud destaca por su baja latencia (5–10 segundos en promedio), soporte para contextos masivos (más de 1 millón de tokens en versiones avanzadas), escalabilidad inmediata y alta disponibilidad gracias a la infraestructura distribuida del proveedor. Además, libera al usuario de tareas de mantenimiento, facilitando la integración rápida en entornos de producción. No obstante, su uso implica costos variables por volumen de consultas, dependencia de un proveedor externo —con el riesgo de cambios en precios o políticas de servicio— y posibles preocupaciones de privacidad, ya que el procesamiento se realiza en servidores de terceros.

La elección entre un modelo local y una API en la nube dependerá del equilibrio entre control de datos, latencia, costo operativo y escalabilidad que se requiera para cada caso de uso. En entornos donde la confidencialidad y el control son prioritarios, un despliegue local puede resultar más ventajoso; mientras que para escenarios que demanden alta capacidad de procesamiento inmediato y flexibilidad de escalado, la nube puede ofrecer beneficios superiores. La Tabla 1 resume las principales diferencias entre ambas opciones:

Tabla 1. Resumen de comparación Qwen-8B local vs Gemini API (Google) *[30]*

Característica	**Qwen-8B (local, LM Studio)**	**Gemini API (Google Cloud)**

Tamaño del modelo	~8 mil M de parámetros (modelo mediano)	Desde 10B hasta cientos de miles de M (Gemini 2.5 Pro)
Infraestructura	Servidor propio (GPU 8 GB)	Infraestructura cloud (Google)
Latencia de respuesta	15–40 s (vectorización + generación)	5–10 s (inferencia optimizada cloud)
Capacidad de contexto	32k tokens (ampliable a 128k)	>60k tokens; hasta 2M tokens en versiones avanzadas
Ajuste de parámetros	Totalmente personalizable en entorno local	Personalizable vía API (temperatura, longitud máx.)
Costo por consulta	Marginal cero tras inversión en hardware	Pago por uso (free tier y luego por tokens consumidos)
Privacidad de datos	Datos permanecen localmente (control total)	Datos enviados a la nube, sujeto a políticas de Google
Disponibilidad	Depende del servidor propio	Alta disponibilidad

5. Conclusión y trabajo futuro

Este trabajo implementa y valida dos flujos agentivos sobre n8n: un chatbot jurídico con RAG (Whisper + Qwen-8B local + Qdrant) que ofrece respuestas contextualizadas a partir de normativa panameña, y un flujo OSINT (WiGLE + MCP) que automatiza consultas geolocalizadas y envía notificaciones por correo.

La principal contribución es una arquitectura modular, reproducible y de bajo costo que, al integrar estándares abiertos como MCP, permite orquestar inteligencia artificial aplicada tanto a la investigación de ciberdelitos como al análisis de redes inalámbricas. Entre las ventajas destacan la trazabilidad de respuestas jurídicas mediante RAG, la baja latencia y el formato tabular en el flujo OSINT, así como la facilidad para intercambiar canales de comunicación (p. ej., Gmail ↔ Telegram) sin alterar la lógica central. Sin embargo, se identificaron limitaciones como la dependencia del corpus legal —que puede generar alucinaciones cuando falta evidencia normativa—, la variabilidad de las APIs externas y la necesidad de infraestructura y mantenimiento en modelos locales, frente a la dependencia y consideraciones de privacidad que implica el uso de soluciones en la nube.

Los resultados obtenidos (Fig. 3 y Fig. 4) sustentan aplicaciones directas en chatbots legales con recuperación de normativa y pipelines OSINT con salida estructurada y envío automático, transferibles a entornos educativos, clínico-forenses y operativos donde la rapidez, la auditabilidad y el control de datos son esenciales.

Como trabajo futuro, se plantea ampliar los flujos OSINT con APIs como VirusTotal, Traccar, Wazuh, herramientas de web scraping y bases de datos como Supabase o Airtable, así como incorporar consultas avanzadas en WiGLE para detectar dispositivos Bluetooth por área geográfica [25].

En el chatbot RAG, se buscará enriquecer el corpus legal con nuevas normativas y material educativo en ciberseguridad, mientras que, a nivel de modelado, se evaluará el uso de LLM de mayor tamaño como gpt-oss-20b o DeepSeek-R1-14B, y SLM de baja latencia para equilibrar costo computacional, calidad y velocidad [20]. Todas las mejoras se someterán a métricas de impacto práctico y se documentarán para su difusión en la comunidad científica, fomentando la adopción responsable de IA en ciberseguridad y análisis jurídico.

En conjunto, este trabajo demuestra que es posible desarrollar soluciones de IA prácticas, seguras y escalables, adaptadas a contextos locales, y establece un marco metodológico y técnico replicable que puede servir de referencia en la convergencia entre automatización, inteligencia artificial y ciberseguridad [31], [33].

Agradecimientos

Los autores desean expresar su agradecimiento a las instituciones y personas que hicieron posible el desarrollo de este trabajo. En particular, se reconoce el apoyo brindado por la universidad y sus departamentos de investigación, así como la colaboración de colegas y expertos que contribuyeron con sus conocimientos y observaciones. Asimismo, se agradece a las comunidades de software libre y de investigación que desarrollan y mantienen las herramientas utilizadas en este estudio, cuyo aporte resulta fundamental para la innovación y el avance científico.

Referencias

[1] Y. Gao, “Retrieval-Augmented Generation for Large Language Models: A Survey,” arXiv preprint arXiv:2312.10997, 2023.

[2] Y. Li et al., “ARM-RAG: Auxiliary Rationale Memory for Retrieval-Augmented Generation,” arXiv preprint arXiv:2311.04177, 2023.

[3] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Advances in Neural Information Processing Systems (NeurIPS), 2020.

[4] M. Li et al., “Corrective Retrieval Augmented Generation (CRAG),” arXiv preprint arXiv:2401.15884, 2024.

[5] A. Asai et al., “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection,” arXiv preprint arXiv:2310.11511, 2023.

[6] X. Zhang et al., “Leveraging Large Language Models for Open-Source Intelligence Analysis,” IEEE Access, vol. 12, pp. 125034–125049, 2024.

[7] B. Clarke and T. Smith, “Implications of Large Language Models for Open-Source Intelligence,” in Proc. European Conf. on Cyber Warfare and Security (ECCWS), 2024.

[8] J. Lee, H. Kim, and J. Park, “Advances and Challenges in Retrieval-Augmented Generation: A Comprehensive Review,” IEEE Access, vol. 12, pp. 115970–115992, 2024.

[9] C. Breitinger, D. Fischer, and J. Garfinkel, “Open-Source Automation Platforms for Digital Investigations: A Performance and Security Analysis,” in Proc. Digital Forensics Research Conf. (DFRWS), 2023, pp. 45–56.

[10] A. K. Singh, S. Sharma, and R. Buyya, “Privacy-Aware Hybrid Cloud Architectures for AI Applications,” Future Generation Computer Systems, vol. 152, pp. 40–54, 2024.

[11] M. Abadi, T. Charton, and D. G. Murray, “Designing Scalable AI Pipelines for Edge and Cloud Integration,” in Proc. IEEE Int. Conf. Cloud Engineering (IC2E), 2022, pp. 211–220.

[12] A. Aletras, D. Tsarapatsanis, V. Preoţiuc-Pietro, and T. Lampos, “Predicting Judicial Decisions of the European Court of Human Rights: A Natural Language Processing Perspective,” PeerJ Computer Science, vol. 3, p. e93, 2016.

[13] M. Catarino, J. Dias, and J. Henriques, “Natural Language Processing in Legal Tech,” in Legal Tech and the Future of Civil Justice, Cambridge University Press, 2023, pp. 221–245.

[14] OpenAI, “openai/whisper: Robust Speech Recognition via Large-Scale Weak Supervision,” GitHub repository, 2023. [Online]. Available: https://github.com/openai/whisper

[15] Qdrant, “RAG Use Case: Advanced Vector Search for AI Applications,” 2025. [Online]. Available: https://qdrant.tech/rag/

[16] Langflow, “Langflow – Build AI Agents Visually,” 2025. [Online]. Available: https://www.langflow.org/

[17] Model Context Protocol, “Introduction to MCP,” 2025. [Online]. Available: https://modelcontextprotocol.io/introduction

[18] WiGLE, “WiGLE: Wireless Network Mapping,” 2025. [Online]. Available: https://wigle.net/index

[19] Qwen Team, “Qwen – Large Language Model Family (Organization page),” Hugging Face, 2024. [Online]. Available: https://huggingface.co/Qwen

[20] P. Belcak *et al.*, “Small Language Models Are the Future of Agentic AI,” *arXiv preprint* arXiv:2506.02153 v1, Jun. 2025. [Online]. Available: https://arxiv.org/abs/2506.02153

[21] Ministerio Público de Panamá, “Denuncia del Delito Informático,” Procuraduría General de la Nación, 2024. [Online]. Available: https://ministeriopublico.gob.pa/denuncias/denuncia-del-delito-informatico/

[22] Autoridad Nacional de Transparencia y Acceso a la Información (ANTAI), “Reglamentan Ley 81 de Protección de Datos Personales,” 2021. [Online]. Available: https://www.antai.gob.pa/reglamentan-ley-81-de-proteccion-de-datos-personales/

[23] Presidencia de la República de Panamá, “Presidente Mulino sanciona leyes que fortalecen el marco legal contra la ciberdelincuencia,” 5 de agosto de 2025. [Online]. Disponible: https://www.presidencia.gob.pa/publicacion/presidente-mulino-sanciona-leyes-que-fortalecen-el-marco-legal-contra-la-ciberdelincuencia

[24] Panamá Ciber Segura, “Legislación Nacional y Organismos Relacionados con Ciberseguridad,” 2025. [Online]. Available: https://panamacibersegura.gob.pa/index.php/legislacion/

[25] OTW, “OSINT: Tracking the Suspect’s Precise Location Using WiGLE.net,” *Hackers-Arise* Blog, 11-Dec-2023. [Online]. Available: https://hackers-arise.com/osint-tracking-the-suspects-precise-location-using-wigle-net/

[26] n8n, “n8n – Powerful Workflow Automation Software & Tools,” 2025. [Online]. Available: https://n8n.io/

[28] LM Studio, “LM Studio – Discover, Download, and Run Local LLMs,” 2025. [Online]. Available: https://lmstudio.ai/

[29] Telegram, “Bots: An Introduction for Developers,” Telegram APIs, 2025. [Online]. Available: https://core.telegram.org/bots

[30] Google, “Gemini API – Documentation (Español-Latinoamérica),” Google AI for Developers, 2025. [Online]. Available: https://ai.google.dev/gemini-api/docs?hl=es-419

[31] B. Singer, K. Lucas, L. Adiga, M. Jain, L. Bauer y V. Sekar, “On the Feasibility of Using LLMs to Autonomously Execute Multi-host Network Attacks,” *arXiv*, preprint arXiv:2501.16466, mayo 2025. [Online]. Disponible: https://arxiv.org/pdf/2501.16466.

[32] Meta Platforms, “WhatsApp Business Platform – Developer Console & Docs,” 2025. [Online]. Available: https://developers.facebook.com/docs/whatsapp/

[33] Anthropic, “Building Effective Agents,” 18-Dec-2024. [Online]. Available: https://www.anthropic.com/research/building-effective-agents