Implementación de Workflows en n8n para Chatbot Jurídico RAG y OSINT WiGLE

Gilberto Ramos[1], Muriel Jaramillo[2], Agustín Sánchez[3], y Abner Ballesteros[4]
[1,2,3,4] Universidad Tecnológica de Panamá, Centro Regional Veraguas, gilberto.ramos1@utp.ac.pa, muriel.jaramillo@utp.ac.pa, agustin.sanchez2@utp.ac.pa, Abner.ballesteros@utp.ac.pa
Profesora Asesora: Vanessa Núñez, Ingeniera de Sistemas y Computación, Universidad Tecnológica de Panamá, Centro Regional Veraguas, vanessa.nunez1@utp.ac.pa

Resumen*– Se presenta la implementación de dos flujos de trabajo con agentes de Inteligencia Artificial (IA) en la plataforma de automatización n8n. El primer flujo corresponde a un chatbot jurídico basado en Retrieval Augmented Generation (RAG), capaz de responder consultas legales específicas utilizando documentos legales indexados en una base vectorial y un modelo de lenguaje grande (LLM) local. El segundo flujo es un agente de OSINT (inteligencia de fuentes abiertas) que integra herramientas externas (WiGLE y SerpAPI) para recopilar y analizar información de redes inalámbricas y búsquedas web en tiempo real. Ambos casos demuestran cómo n8n permite orquestar modelos de IA con fuentes de datos externas de forma visual y sin código extenso, preservando el rigor técnico. Se emplearon tecnologías abiertas como el modelo Qwen-8B ejecutado localmente y modelos en la nube como gemini-2.0-flash, comparando su desempeño. Además, se incorporó reconocimiento de voz automático (Whisper) para entrada de voz al chatbot. Los resultados muestran que es factible construir soluciones especializadas de IA con n8n que combinan modelos locales y servicios externos, manteniendo precisión y flexibilidad. Se discuten las diferencias entre utilizar un LLM local frente a uno en la nube, así como consideraciones de rendimiento. Finalmente, se propone la futura integración de herramientas adicionales (VirusTotal, Supabase, Traccar) para ampliar las capacidades de la plataforma en ciberseguridad, gestión de datos y seguimiento geográfico.

***Palabras claves*—APIs, Chatbot, IA, LLM, n8n, OSINT, Qdrant, Qwen-8B, RAG, Tokens, WiGLE, Whisper.**

I. Introducción

En los últimos años ha surgido la posibilidad de crear **agentes de IA** (no limitados a simples chatbots o a IA generativa) combinando modelos de lenguaje con diversas fuentes de información. La técnica de **Retrieval Augmented Generation (RAG)** es, en realidad, un enfoque de **expansión de contexto**: permite a cualquier agente de IA —sea un asistente conversacional, un buscador semántico o una herramienta de análisis— aprovechar conocimientos externos indexados en una base vectorial para generar respuestas más precisas y fundamentadas [1]. Un agente basado en RAG puede, por ejemplo, conectar con documentos internos, datos en bases SQL o APIs especializadas, recuperar los fragmentos más relevantes y luego emplear un LLM para elaborar una salida contextualizada. N8n es una plataforma de automatización de flujos de trabajo open source que conecta cientos de servicios, herramientas y APIs mediante una interfaz visual y lógica condicional, sin necesidad de conocimientos avanzados de programación. Esto la hace ideal para implementar agentes de IA multi-paso que integren diversas fuentes de datos y modelos en un solo flujo.. De hecho, Anthropic distingue entre *workflows* (flujos predefinidos orquestados por código) y *agentes* (sistemas donde el LLM dirige dinámicamente el proceso), recomendando utilizar patrones simples y componibles para la mayoría de las aplicaciones en lugar de agentes autónomos complejos [9]. Siguiendo estos principios, los flujos aquí presentados emplean LLMs aumentados con herramientas (búsqueda de contexto, APIs externas) en vez de agentes totalmente libres. Cabe mencionar que el bloque fundamental de un sistema agéntico es un LLM enriquecido con capacidades como recuperación de datos, uso de herramientas y memoria externa. En esa línea, Anthropic recientemente introdujo *Model Context Protocol* (MCP), ver Figura 1, un protocolo abierto que estandariza la conexión de modelos con distintas fuentes de datos y herramientas – un "USB-C" para aplicaciones de IA [11], [31].

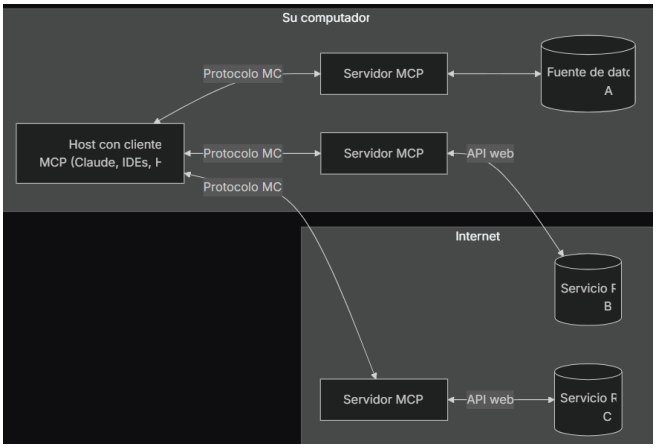

Figura 1 - MCP sigue una arquitectura cliente-servidor en la que una aplicación host puede conectarse a varios servidores

En este trabajo se implementó un **chatbot jurídico RAG** en n8n, diseñado para responder preguntas legales apoyándose en una base de datos de textos jurídicos proporcionados. Asimismo, se desarrolló un flujo de **OSINT** para automatizar la obtención de información abierta. El agente OSINT utiliza la API de **WiGLE** [16], un servicio en línea con una amplia base de datos colaborativa de redes inalámbricas

geolocalizadas, y la herramienta **SerpAPI** integrada en n8n para realizar búsquedas web en Google de forma automatizada. De este modo, un agente puede combinar datos técnicos —por ejemplo, la ubicación de un punto de acceso Wi-Fi — con información contextual obtenida en tiempo real de internet.

Estos casos demuestran cómo, sin código complejo, n8n permite combinar IA generativa con fuentes externas para dos fines: en el ámbito legal, extraer con precisión normativas; y en OSINT, buscar datos públicos. Se aprovecha su arquitectura modular y el nodo de agente. Luego se detallan método, resultados y comparan modelos locales (Qwen-8B) versus APIs en la nube (Gemini), además de evaluar el valor educativo y proponer mejoras futuras.

II. Metodología

La metodología se divide en dos partes correspondientes a cada flujo implementado. A continuación, se describen en detalle los componentes y pasos del Flujo 1: Chatbot RAG para ciberdelito, ver Figura 2, y del Flujo 2: Agente OSINT con WiGLE y SerpAPI, incluyendo las herramientas de software empleadas, ver Figura 3.

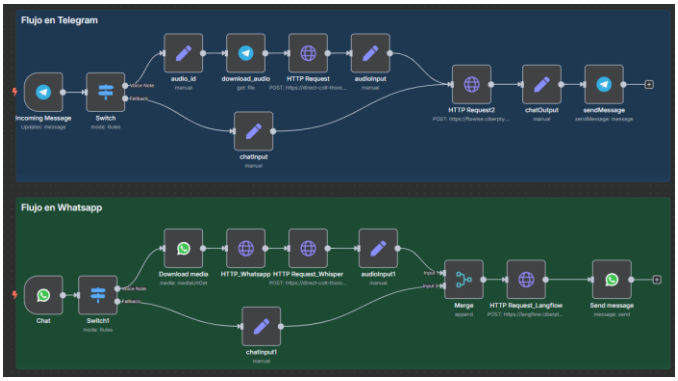

Figura 2 - Chatbot RAG para ciberdelito con Telegram y Whatsapp

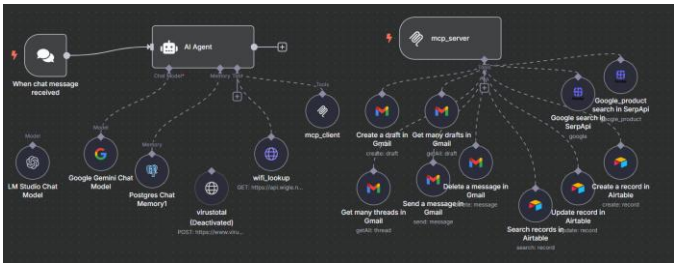

Figura 3 - Agente OSINT con WiGLE y conectado al MCP server que incluye las tools de SerAPI, Airtable y Gmail

Flujo 1: Chatbot RAG de orientación sobre ciberdelitos

Objetivo y panorama general: El chatbot RAG busca brindar respuestas jurídicas preliminares a consultas sobre delitos informáticos, en lenguaje natural y en menos de 40 segundos, apoyándose en la normativa panameña (e.g. Ley 81/2019) y jurisprudencia relevante [24]. El agente atiende a través de mensajería instantánea (se integró un bot de Telegram y un endpoint de WhatsApp Business en n8n), permitiendo que los usuarios realicen consultas por texto o voz. Cuando el usuario envía una pregunta (por ejemplo, *"¿Qué sanciones establece la Ley 81/2019 por filtración de datos?"*), el flujo realiza una búsqueda de contexto en un corpus legal y genera una respuesta elaborada con el LLM, devolviéndola por el mismo canal al usuario.

Arquitectura del flujo: El Flujo 1 se compone de cuatro etapas principales: (1) disparador multicanal, (2) preprocesamiento (voz a texto), (3) motor RAG (vectorización + LLM) y (4) publicación de la respuesta. A continuación, se detallan estos componentes:

1. **Disparador multicanal:** Se configuraron nodos disparadores en n8n para integrar dos canales de mensajería: un nodo *Telegram Trigger* (usando la API de bots de Telegram) y un webhook de *WhatsApp Cloud API* (canal empresarial de Meta) [37], [29]. Esto permite que cada nuevo mensaje en Telegram o WhatsApp active el flujo en n8n de forma inmediata.
2. **Preprocesamiento de voz:** Si el usuario envía una nota de voz o audio, el flujo invoca un paso de transcripción usando Whisper, el modelo de reconocimiento automático de voz de código abierto de OpenAI, ver Figura 4. Whisper fue elegido por su capacidad robusta para convertir audio multilingüe a texto de forma rápida. El audio se descarga mediante n8n y se envía a un servicio local de Whisper (Endpoint en Ubuntu) que retorna la transcripción del mensaje en texto.

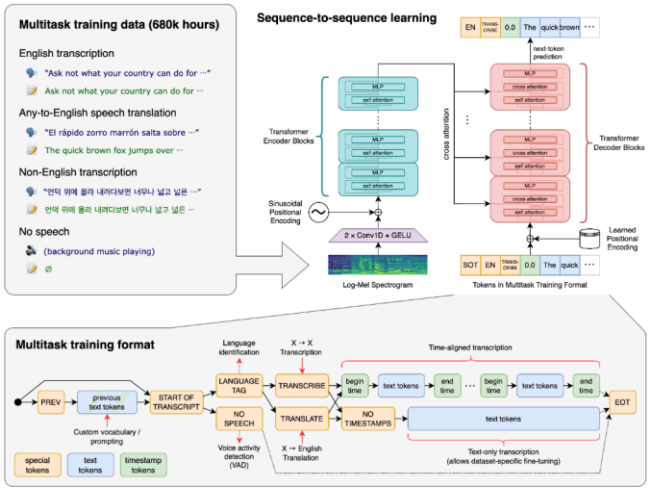

Figura 4 - *Whisper* es un modelo multitarea de reconocimiento de voz entrenado con un vasto conjunto de audios diversos, capaz de transcribir en múltiples idiomas, traducir y detectar automáticamente el idioma [18].

3. **Recuperación Aumentada con Generación (RAG):** Esta es la etapa central del agente, donde se combina la

recuperación de información con la generación de lenguaje natural:

- **Vectorización de documentos legales:** Se preparó un corpus con la Ley 81/2019 y jurisprudencia relevante en materia de ciberdelitos [23], [25], [26], [27]. Cada documento fue preprocesado y dividido en fragmentos (chunks) de tamaño controlado (chunk_size = 1 000 tokens, chunk_overlap = 200 tokens) para mantener la coherencia contextual. A continuación, cada chunk se transformó en un vector de incrustación (embedding) mediante el modelo text-embedding-qwen3-embedding-4b. Estas representaciones vectoriales, que capturan la similitud semántica de cada fragmento, permiten indexar y recuperar con precisión los pasajes más relevantes durante la fase de recuperación, ver Figura 5.

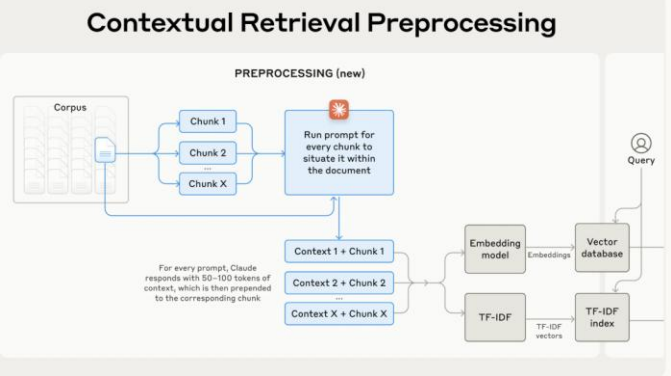

Figura 5 - *La recuperación contextual es una técnica de preprocesamiento que mejora la precisión de la recuperación [10].*

- **Base de datos vectorial (Qdrant):** Todos los embeddings generados con el modelo **text-embedding-qwen3-embedding-4b** se almacenaron en Qdrant, un motor de búsqueda vectorial de alto desempeño escrito en Rust, ver Figura 6, [33]. Qdrant actúa como base vectorial, permitiendo realizar búsquedas por similitud: dado un vector de consulta (el embedding de la pregunta del usuario), retorna los documentos más relevantes semánticamente. Ofrece un servicio listo para producción con una API conveniente para almacenar y buscar vectores, incluyendo filtros por metadatos cuando es necesario. Esta arquitectura garantiza que el chatbot pueda recuperar rápidamente párrafos legales pertinentes a la consulta del usuario.

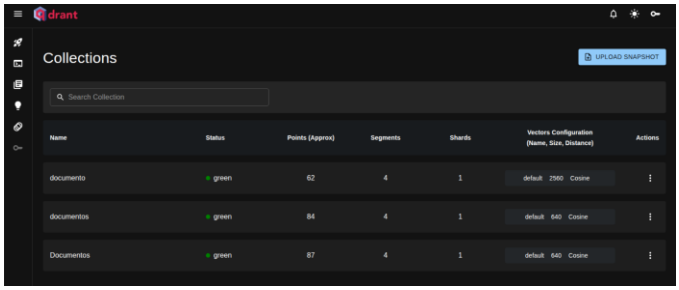

Figura 6 - Qdrant es un motor de base de datos vectorial de alto rendimiento, y ya tenemos cargados nuestros vectores (embeddings) en la colección correspondiente.

- **LLM local (Qwen-8B en LM Studio):** Para la generación de la respuesta final se utilizó un modelo de lenguaje grande local llamado Qwen-8B, desplegado a través de la herramienta LM Studio. Qwen-8B es un modelo de ~8 mil millones de parámetros, perteneciente a la familia Qwen (desarrollada abiertamente por Alibaba Cloud), y soporta un contexto amplio (hasta ~32k tokens por defecto) [20]. LM Studio provee un entorno unificado para descargar y ejecutar localmente modelos LLM de forma optimizada en hardware personal [36]. En este flujo, Qwen-8B se ejecutó en una laptop (GPU de 8 GB y 32 de RAM) y expuso un endpoint usando LM Studio, ver Figura 7. Para cada consulta de usuario, primero se genera su embedding y se consulta Qdrant; luego, los textos legales más relevantes obtenidos se concatenan a la pregunta del usuario como contexto (prompt extendido) y se envían al modelo Qwen, que elabora una respuesta fundamentada con esa información. Este patrón RAG permite que el LLM no dependa solo de su conocimiento entrenado, sino que esté **aumentado** con información recuperada específicamente del dominio legal, mejorando la precisión y actualidad de las respuestas.

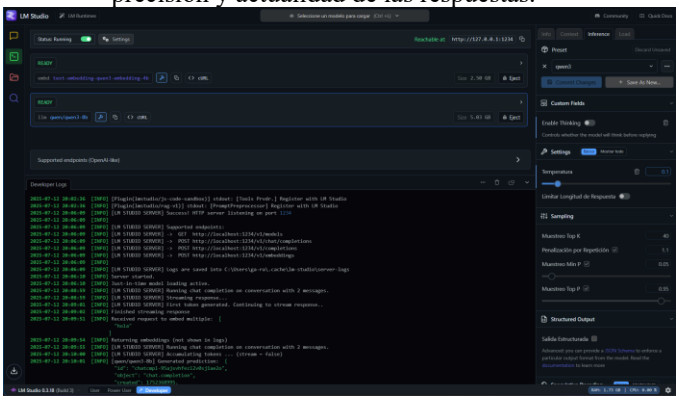

Figura 7 - LM Studio es una interfaz local para cargar y servir modelos de lenguaje de forma sencilla.

- **Orquestación con Flowise/Langflow:** Para construir este pipeline RAG de manera visual, se

emplearon las herramientas Flowise y Langflow. **Flowise** es una plataforma open-source que ofrece una interfaz *drag-and-drop* para crear aplicaciones LLM y agentes de IA, proporcionando bloques modulares para componer flujos complejos, ver Figura 8, [5]. **Langflow**, por su parte, es un framework visual intuitivo para construir aplicaciones con múltiples agentes y RAG; es completamente de código abierto, compatible con diversos LLMs y almacenes vectoriales, y permite desplegar agentes o servidores MCP fácilmente, ver Figura 9, [38]. Su interfaz de arrastrar y soltar facilita la creación rápida de cadenas de llamadas a modelos y herramientas sin necesidad de programar en detalle. Langflow permite configurar parámetros del modelo (p. ej., temperatura, top-p) y exponer el flujo como un servicio REST local. De este modo, n8n interactúa con la lógica RAG a través de un nodo HTTP Request, invocando el endpoint REST de Langflow/Flowise cada vez que llega una consulta.

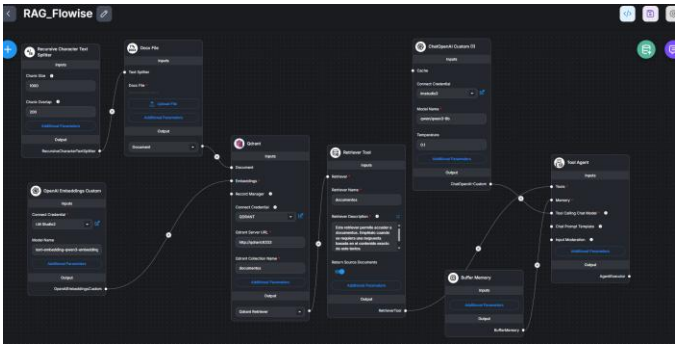

Figura 8 - Flowise es una plataforma de orquestación basada en nodos para diseñar pipelines de RAG; aquí se muestra uno de los flujos RAG que implementamos.

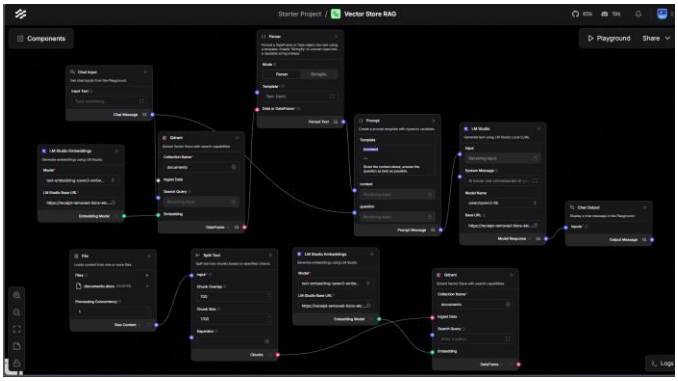

Figura 9 - En esta vista se muestra el flujo RAG en Langflow que genera embeddings, los indexa y recupera de Qdrant, y lo exponemos como endpoint HTTP para ser consumido desde n8n.

4. **Publicación de la respuesta:** Finalmente, la respuesta generada por el modelo Qwen-8B se envía de vuelta al usuario a través del mismo canal de origen. En el caso de Telegram, se utiliza el nodo propio de *Telegram Send Message* de n8n para mandar la respuesta al chat correspondiente. Para WhatsApp, se invoca la API de WhatsApp Cloud mediante un nodo HTTP (con credenciales preconfiguradas) para enviar el mensaje de respuesta.

Durante el desarrollo del Flujo 1 se realizaron ajustes de parámetros para garantizar un buen desempeño. Por ejemplo, se calibró la temperatura y *top-p* del modelo Qwen-8B para obtener un tono formal y preciso (en consultas legales se prefieren temperaturas bajas para mayor determinismo). También se limitó el máximo de tokens en las respuestas para evitar excesos, y se optimizó la selección de contexto desde Qdrant (trayendo solo los 3–5 fragmentos más relevantes por consulta). Tras implementar el flujo completo, se llevó a cabo un piloto con usuarios internos, midiendo tiempos de respuesta, exactitud percibida y retroalimentación para iterar mejoras. En resumen, el Flujo 1 demuestra cómo integrar un LLM local con recuperación de conocimiento específica (RAG) en un asistente jurídico. La combinación de n8n con herramientas especializadas (Whisper, Qdrant, Langflow) permitió construir un chatbot legal a la medida, preservando el control sobre los datos (todo el procesamiento ocurre on-premise) y cumpliendo los objetivos de rapidez, concisión y pertinencia en las respuestas al usuario final.

Flujo 2: Agente OSINT con WiGLE, SerpAPI y MCP para Gmail

Objetivo y panorama general: El segundo flujo tiene por objetivo agilizar la recolección de inteligencia de fuentes abiertas (OSINT) sobre ciertos artefactos técnicos (particularmente redes Wi-Fi) y entregar un resumen informativo al analista a través de correo electrónico. Un caso de uso típico es: dado un identificador de red inalámbrica (por ejemplo, un BSSID o SSID sospechoso obtenido durante una investigación), obtener su posible ubicación geográfica y datos asociados, además de buscar menciones relevantes en la web, para luego enviar un informe conciso por email con todos los hallazgos [2]. Tradicionalmente, un analista realizaría estas indagaciones manualmente en múltiples plataformas; con este flujo en n8n se automatiza gran parte de ese trabajo, permitiendo obtener resultados consistentes en segundos.

Arquitectura del flujo: El Flujo 2 se compone de los siguientes pasos secuenciales:

- **Inicio (Trigger):** Se utiliza un disparador manual (*Manual Trigger*) en n8n para iniciar el flujo.
- **Consulta a WiGLE:** El primer nodo OSINT es un HTTP Request configurado para llamar a la API REST de **WiGLE**. WiGLE (Wireless Geographic Logging Engine) es una plataforma colaborativa que consolida datos de localización e información de redes inalámbricas a nivel mundial, ver Figura 10. Voluntarios aportan datos mediante *wardriving* (escaneo de redes Wi-Fi con sus coordenadas GPS), y WiGLE ofrece interfaces de consulta a esa base de

datos pública. En nuestro flujo, se emplea el endpoint /api/v2/network/search de WiGLE mediante una solicitud GET autenticada (usando un token API), [41]. Cabe mencionar que la API de WiGLE también soporta consultas por área geográfica: por ejemplo, especificando coordenadas o un rectángulo de búsqueda para listar redes Wi-Fi (filtradas por tipo de seguridad, como WPA2) presentes en esa zona.

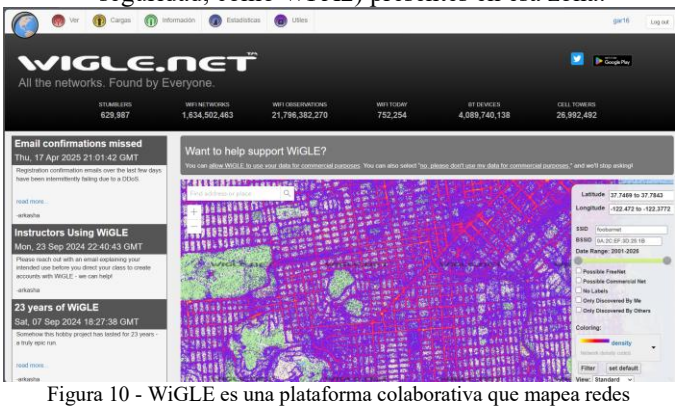

Figura 10 - WiGLE es una plataforma colaborativa que mapea redes inalámbricas descubiertas por usuarios [40].

- **Consulta a SerpAPI (Google):** A continuación, el usuario puede realizar una búsqueda web general para recabar información complementaria. Para ello se utiliza **SerpAPI**, un servicio que provee una API para obtener resultados de Google de forma estructurada, ver Figura 11, [3]. SerpAPI esencialmente resuelve la consulta en Google internamente (maneja proxies, captchas, etc.) y devuelve los resultados parseados en formato JSON.

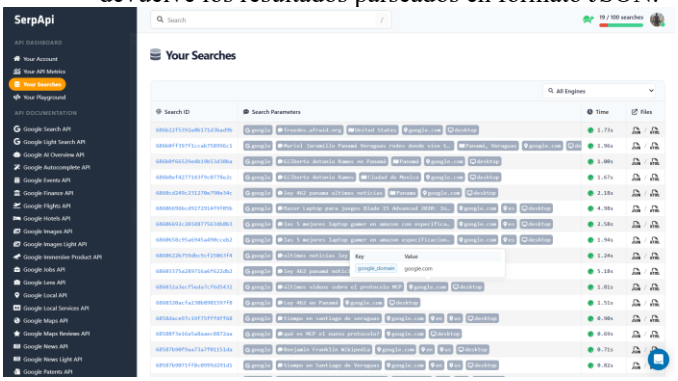

Figura 11 - Se observan los parámetros usados, el tiempo de respuesta y los archivos generados por cada búsqueda en SerAPI.

- **Normalización y resumen:** Los procesos se realizan de forma independiente en n8n: cuando se consulta WiGLE, el nodo extrae la latitud, longitud y fecha del último avistamiento; por otro lado, al consultar SerpAPI se capturan los resultados de búsqueda con sus títulos. A continuación, cada JSON se interpreta por separado y el agente construye un resumen textual conciso basado en el origen de los datos.
- **Envío mediante MCP a Gmail:** Una vez generado el resumen, el agente cuando se le indica procede a enviarlo por correo electrónico. En este caso usamos un pequeño servidor MCP como puente hacia Gmail. En otras palabras, el servidor MCP está configurado con las credenciales seguras de Gmail y actúa bajo demanda cuando n8n le solicita "envía este mensaje al correo X", [21].
- **Finalización:** Tras invocar el envío por MCP, el flujo termina. En cuestión de segundos desde su activación, el analista recibe en su correo Gmail el resumen OSINT generado.

Este agente OSINT demuestra la integración práctica de múltiples fuentes abiertas y servicios web en un solo flujo automatizado, proporcionando al analista un reporte personalizado, reproducible y extensible. La orquestación en n8n y la separación de notificación mediante MCP facilitan futuras ampliaciones, contribuyendo a procesos de investigación digital más eficientes y escalables.

III. RESULTADOS

Desempeño de los flujos implementados

Flujo 1 (Chatbot RAG):

El chatbot jurídico fue probado con consultas reales sobre ciberdelitos y protección de datos personales. El tiempo medio de respuesta fue de 15–40 segundos, cumpliendo la meta de rapidez, incluso incluyendo el paso de transcripción de voz (Whisper) y generación de texto (Qwen-8B). Las respuestas, gracias a la base vectorial, resultaron pertinentes y referenciadas a la normativa panameña, recomendando acciones claras al usuario (como citar artículos de ley o contactar a las autoridades). En ocasiones se detectaron errores en alucinaciones porque el modelo no consultaba la base vectorial antes de responder, aunque la precisión dependió del contenido del corpus cargado y de la pregunta pertinente que haga el usuario. Telegram funcionó de forma fluida y WhatsApp mostró una leve demora extra por la API, pero dentro de márgenes aceptables. El flujo cumplió su objetivo didáctico y práctico, guiando a los usuarios con información legal relevante y recomendaciones concretas.

Flujo 2 (Agente OSINT):

Las pruebas con BSSID y SSID reales demostraron que el flujo OSINT generó y envió resúmenes en aproximadamente 5 segundos. WiGLE respondió en 1–2 segundos, SerpAPI en otros 2, y el correo vía MCP/Gmail llegó de forma inmediata. El sistema fue capaz de ubicar redes inalámbricas correctamente y entregar contexto web relevante (noticias, foros, incidentes), o informar cuando no se hallaban resultados. La arquitectura es fácilmente extensible: pueden

sumarse nodos para otras fuentes o formatos, ampliando los casos de uso más allá de redes Wi-Fi, lo que demuestra la versatilidad del enfoque modular en n8n.

Comparativa: Modelo local vs API en la nube

Se comparó el rendimiento entre el modelo local Qwen-8B (en LM Studio) y la API Gemini de Google. Qwen-8B ofrece bajo costo por uso, control total de datos y buen soporte para contextos largos (~32k tokens), pero requiere infraestructura propia, mayor latencia y mantenimiento manual. Gemini API, ejecutada en la nube de Google, ofrece menor latencia (~5–10 s), contexto masivo (más de 1M tokens), integración directa vía HTTP y actualizaciones automáticas, pero implica costos variables y depender de un proveedor externo, con los riesgos de privacidad que esto conlleva [42]. La motivación fue evaluar compromisos entre rendimiento, costos y facilidad de integración al implementar este tipo de agentes de IA. La **Tabla 1** resume la comparación en varios aspectos clave:

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	15–30 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

Tabla 1. Resumen de comparación Qwen-8B local vs Gemini API (Google):
(Fuente: Elaboración propia basada en la experiencia del proyecto y documentación de Qwen (HuggingFace, LM Studio) y Gemini API (Google Cloud)

En resumen, ambas alternativas son válidas y la elección depende del contexto. El modelo local resulta idóneo donde la privacidad y el control son prioritarios; la nube es preferible para mayor escalabilidad, rapidez y menor carga de mantenimiento. El uso de estándares abiertos como MCP facilita combinar ambos enfoques y permite seleccionar dinámicamente el recurso adecuado según la carga o la naturaleza de la consulta.

IV. DISCUSIÓN

Los resultados muestran que **n8n** es una plataforma robusta para integrar agentes de IA en flujos de trabajo personalizados y prácticos. En el Flujo 1 logramos implementar un chatbot jurídico actualizado y flexible, capaz de procesar consultas (por ejemplo, "Fui víctima de una estafa virtual, ¿qué debo hacer o a dónde puedo poner mi denuncia?", ver Figura 12) y devolver respuestas precisas directamente en WhatsApp sin depender de servicios comerciales externos. En la Figura 13, en n8n se ve cómo el nodo "HTTP Request Langflow" envía esa pregunta a nuestro endpoint local y recibe un JSON con el texto generado, que luego se envía al usuario en su celular. Esto permite a organizaciones o equipos con recursos limitados construir asistentes inteligentes adaptados a su realidad y controlar totalmente sus datos.

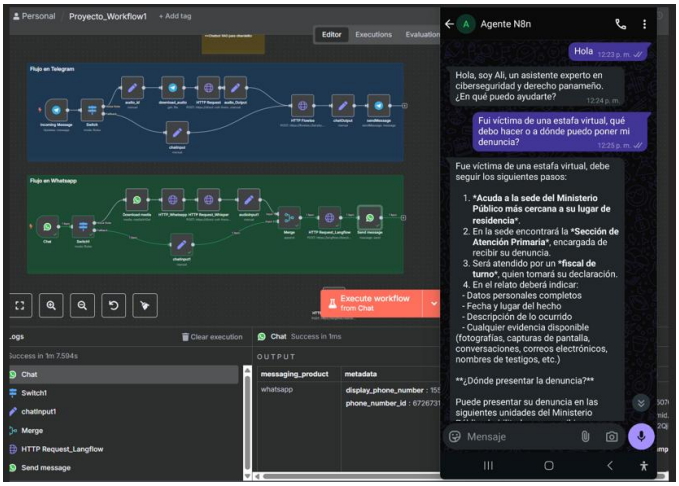

Figura 12 – Se muestra el flujo de WhatsApp en n8n donde el texto de la consulta se envía al nodo HTTP Request (Langflow) y, tras recibir el JSON de respuesta, se entrega al usuario en la pantalla del móvil.

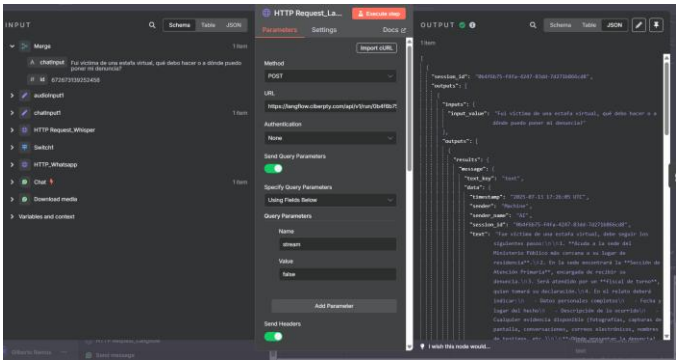

Figura 13 - Captura del nodo HTTP Request de Langflow en n8n: a la izquierda, los parámetros de la solicitud POST al endpoint local; a la derecha, el JSON de salida con la respuesta generada por el modelo.

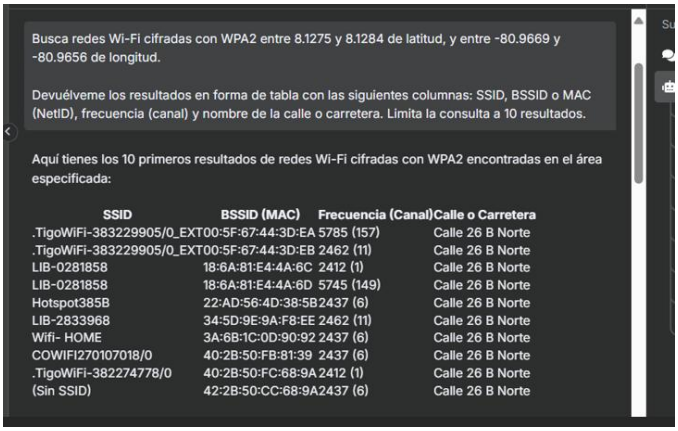

Figura 14 – Ejecución de una consulta y con resultado exitoso, toca señalar que en ocasiones te pide los parámetros necesarios.

La **precisión** de las respuestas, potenciada por la técnica RAG, depende tanto de la calidad del corpus cargado como de la capacidad del modelo. Para despliegues en producción o aplicaciones sensibles, es recomendable incorporar validación humana ("human-in-the-loop") en ciertas etapas. Herramientas como Flowise o Langflow ya soportan pasos de aprobación manual dentro del flujo, facilitando este control. En cuanto a **escalabilidad**, una posible mejora es enrutar preguntas simples a modelos ligeros (o a flujos deterministas) y reservar el modelo local potente para consultas complejas, optimizando recursos y tiempos de respuesta.

En el Flujo 2, la automatización OSINT con n8n y MCP demostró ser versátil y fácilmente extensible. Por ejemplo, en una ejecución reciente el agente buscó redes Wi-Fi cifradas con WPA2 en el área delimitada por latitud 8.1275–8.1284 y longitud –80.9669– –80.9656, devolviendo los 10 primeros resultados en forma de tabla con SSID (por ejemplo, .TigoWiFi-383229905/0_EXT00:5F:67:44:3D:EA), BSSID (MAC), frecuencia (canal) y calle (Calle 26 B Norte), ver Figura 14. Un reto importante es la confiabilidad de las fuentes: WiGLE puede tener datos incompletos y SerpAPI resultados no verificados, por lo que la interpretación final debe ser siempre supervisada por un humano. Integrar modelos LLM externos (como Gemini o GPT-4) para redactar informes más completos o identificar patrones es una mejora viable usando MCP, manteniendo la flexibilidad y modularidad del flujo. Además, en ocasiones es necesario habilitar la memoria del agente, pero debe desactivarse para operaciones puntuales como enviar correos por Gmail o consultar redes vía WiGLE.

Destacamos la **importancia de los estándares abiertos** como MCP, que facilitan la interoperabilidad y la rápida integración de nuevas herramientas, permitiendo intercambiar o sumar componentes (como cambiando el canal de notificación de Gmail a Telegram) sin modificar la lógica central del flujo. Esta arquitectura modular es clave en entornos donde la tecnología avanza rápido y los requisitos pueden cambiar.

La **usabilidad** y la **curva de aprendizaje** de estas plataformas (n8n, Langflow) resultaron accesibles: con conocimientos básicos en APIs, automatización y estructuras como JSON, el equipo pudo ensamblar componentes avanzados. Sin embargo, sigue siendo crucial comprender en profundidad cada herramienta y sus límites. Por ejemplo, es fundamental saber configurar correctamente los embeddings en Qdrant, dominar los métodos **POST** y **GET** en las APIs, y manejar el intercambio de datos en formato JSON entre los diferentes nodos de los flujos [13], [14]. La documentación oficial y los recursos comunitarios fueron vitales para el éxito del proyecto, ya que permitieron resolver dudas específicas y adaptar los flujos a necesidades particulares. Es esencial aplicar buenas prácticas desde el inicio: manejo de errores, validación de las entradas del usuario, registro adecuado de logs y un respeto estricto por la privacidad y la ética en el tratamiento de los datos.

Futuras pruebas y mejoras

Como siguiente paso, planeamos ampliar los flujos con más herramientas OSINT, integrando APIs como **VirusTotal** (para análisis de archivos y URLs) [12], **Traccar** (rastreo GPS en tiempo real) y bases de datos como **Supabase** o **Airtable** para registrar hallazgos de manera estructurada [8], [28], [19]. También se experimentará con consultas basadas en coordenadas geográficas (por ejemplo, listar redes WPA2 por área usando WiGLE). Para el **chatbot RAG**, se prevé enriquecer el corpus legal con más normativas de ciberseguridad y contenidos educativos, e incorporar un

segundo modelo (local o cloud) como respaldo para preguntas fuera del alcance actual, garantizando cobertura total.

Finalmente, se evaluará la incorporación de modelos LLM más grandes (como Qwen3-14B o DeepSeek-R1-14B), analizando el balance entre costo computacional y calidad de respuesta [43]. Además, exploraremos el uso de **Small Language Models (SLM)**, que, según Belcak et al., son inherentemente más económicos y adecuados para tareas repetitivas y especializadas en sistemas agentivos, al ofrecer inferencia de baja latencia en hardware convencional [22]; asimismo, Splunk señala que los SLM están optimizados para precisión en dominios concretos y operan con recursos significativamente menores que los LLM, sin sacrificar efectividad [17]. Todas estas pruebas se enfocarán en medir el impacto práctico de cada mejora, compartiendo los resultados con la comunidad para fomentar la adopción responsable de IA aplicada en ciberseguridad y análisis jurídico.

V. CONCLUSIÓN

Este trabajo presentó la implementación de dos flujos de agentes de IA sobre la plataforma n8n, demostrando cómo combinar herramientas de inteligencia artificial abiertas con servicios web existentes para tareas específicas en los campos de ciberdelito y OSINT. El chatbot legal basado en RAG ofreció respuestas contextualizadas usando legislación panameña, integrando tecnologías como Whisper para transcripción de voz, Qwen-8B como LLM local para generación de texto, y Qdrant como base vectorial para recuperar información relevante. Por otro lado, el agente OSINT automatizó la recolección y entrega de inteligencia sobre redes Wi-Fi y búsquedas web, enlazando WiGLE y Google (vía SerpAPI) con notificaciones por correo mediante MCP, sin requerir intervención manual, solo la petición que haga el usuario desde n8n.

La experiencia deja varios hallazgos relevantes: Primero, n8n se consolida como una plataforma eficaz para orquestar flujos de IA multipaso, facilitando la integración de componentes diversos en un solo proceso coherente. Segundo, los modelos locales permiten independencia de la nube y control de datos, aunque exigen infraestructura propia y gestión técnica; en nuestro caso, un LLM local potenciado con RAG fue suficiente para un chatbot jurídico funcional. Tercero, los servicios externos (APIs) complementan y potencian estos flujos, aportando datos y capacidades que sería costoso implementar desde cero; la combinación de fuentes open-source y servicios cloud, facilitada por estándares como MCP, permite construir soluciones flexibles y adaptables.

Finalmente, la comparación entre modelo local y servicio cloud (como Gemini API) muestra que la mejor opción depende del contexto: volumen de uso, sensibilidad de los datos y recursos disponibles. La arquitectura modular desarrollada permite intercambiar o combinar ambos enfoques según se requiera, lo que es una buena práctica para futuros proyectos.

En suma, este proyecto demuestra que es posible crear soluciones de IA prácticas y de bajo costo, adaptadas a necesidades locales de ciberseguridad y análisis, aprovechando la sinergia entre herramientas abiertas y plataformas de automatización visual. El enfoque propuesto puede servir de base para desarrollos futuros en ciberinvestigación y protección de datos, contribuyendo al avance de agentes de IA efectivos, seguros y accesibles en distintos sectores.

REFERENCIAS

[1] M. Farcas, “Build a custom knowledge RAG chatbot using n8n,” n8n Blog, Jan. 2025. [Online]. Disponible: blog.n8n.io

[2] C. Fragoso, “Integrating Wireless Data into Your OSINT Investigations,” Maltego Blog, 17-Jul-2023. [Online]. Available: https://www.maltego.com/blog/integrating-wireless-data-into-your-osint-investigations/

[3] A. Barron, “How to Build an AI Agent with n8n and Live Google Search Data,” SerpApi Blog, 08-Feb-2025. [Online]. Available: https://serpapi.com/blog/how-to-build-an-ai-agent-with-n8n-and-live-google-search-data/

[4] A. Alford, “OpenAI Releases 1.6 B Parameter Multilingual Speech Recognition AI Whisper,” InfoQ, 04-Oct-2022. [Online]. Available: https://www.infoq.com/news/2022/10/openai-whisper-speech/

[5] Flowise AI, “Flowise – Build AI Agents Visually,” 2025. [Online]. Available: https://flowiseai.com/

[6] Camelor Lab, “Langflow Documentation – Welcome to Langflow,” 2025. [Online]. Available: https://docs.langflow.org/

[7] Qdrant, “What is Qdrant? – Documentation,” 2023. [Online]. Available: https://qdrant.tech/documentation/

[8] A. Tananaev, “Traccar GPS Tracking System – Overview,” GitHub, 2025. [Online]. Available: https://github.com/traccar/traccar

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

[10] Anthropic, “Introducing Contextual Retrieval,” 19-Sep-2024. [Online]. Available: https://www.anthropic.com/news/contextual-retrieval

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

[12] VirusTotal, “VirusTotal – Official Site,” 2025. [Online]. Available: https://www.virustotal.com/

[13] MDN Web Docs, “HTTP request methods: GET,” 2025. [Online]. Available: https://developer.mozilla.org/docs/Web/HTTP/Methods/GET

[14] MDN Web Docs, “HTTP request methods: POST,” 2025. [Online]. Available: https://developer.mozilla.org/docs/Web/HTTP/Methods/POST

[15] n8n Docs, “HTTP Request Node,” 2025. [Online]. Available: https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.httprequest/

[16] WiGLE, "WiGLE API Overview (v2)," 2025. [Online]. Available: https://api.wigle.net/

[17] Splunk, "LLMs vs. SLMs: The Differences in Large & Small Language Models," 12-Mar-2025. [Online]. Available: https://www.splunk.com/en_us/blog/learn/language-models-slm-vs-llm.html

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

[19] Airtable, "AI-Native Platform for Building Enterprise Apps," Airtable, 2025. [Online]. Available: https://www.airtable.com/

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

[21] n8n, "Gmail Credentials – Built-in Send Email (Gmail)," n8n Docs, 2025. [Online]. Available: https://docs.n8n.io/integrations/builtin/credentials/sendemail/gmail/

[22] 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

[23] 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/

[24] 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/

[25] *La Estrella de Panamá*, "Seguridad Cibernética (Sección Temática)," 2024. [Online]. Available: https://www.laestrella.com.pa/tag/-/meta/seguridad-cibernetica

[26] S. M. González, "Delitos Informáticos en Panamá: una amenaza que aumentó 113 % en los últimos meses," *TVN-2 Contenido Exclusivo*, 23 Sep 2024. [Online]. Available: https://www.tvn-2.com/contenido-exclusivo/delitos-informaticos-panama-ciberseguridad-delincuentes-proyecto-de-ley-ministerio-publico-investigacion-phishing_1_2157643.html

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

[28] Supabase, "Supabase – The Postgres Development Platform," 2025. [Online]. Available: https://supabase.com/

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

[30] 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/

[31] Model Context Protocol, "modelcontextprotocol/servers – Reference Implementations," GitHub repository, 2025. [Online]. Available: https://github.com/modelcontextprotocol/servers

[32] Google, "Google AI Studio – Get an API Key," 2025. [Online]. Available: https://aistudio.google.com/apikey

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

[34] n8n, "n8n – Powerful Workflow Automation Software & Tools," 2025. [Online]. Available: https://n8n.io/

[35] n8n Docs, "Advanced AI Example: Understand Agents," 2025. [Online]. Available: https://docs.n8n.io/advanced-ai/examples/understand-agents/

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

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

[38] Langflow, "Langflow – Build AI Agents Visually," 2025. [Online]. Available: https://www.langflow.org/

[39] LangChain, "LangChain – The Platform for Reliable Agents," 2025. [Online]. Available: https://www.langchain.com/

[40] WiGLE, "WiGLE: Wireless Network Mapping," 2025. [Online]. Available: https://wigle.net/index

[41] WiGLE, "WiGLE API v2 – Network Search and Information Tools," 2025. [Online]. Available: https://api.wigle.net/swagger#/Network%20search%20and%20information%20tools/search_2

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

[43] Ollama, "Ollama Search – Explore and run open LLM models locally," 2025. [Online]. Available: https://ollama.com/search