

Despliegue de n8n en producción para automatización de flujos de trabajo

Deployment of n8n in production for workflow automation

Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Ibarra García, E.P. y Cruz Estrada, J.C. (2026). Despliegue de n8n en producción para automatización de flujos de trabajo. *Cuadernos Técnicos Universitarios de la DGTIC*, 4(2), páginas (8 - 28). <https://doi.org/10.22201/dgtic.30618096e.2026.4.2.171>

Ernesto Pathros Ibarra García

Instituto de Investigaciones Sociales
Universidad Nacional Autónoma de México
pathros@sociales.unam.mx
ORCID: 0009-0007-5972-5558

Julio Cesar Cruz Estrada

Instituto de Investigaciones Sociales
Universidad Nacional Autónoma de México
juliocruz@sociales.unam.mx
ORCID: 0000-0003-0107-6616

Resumen

La automatización de flujos de trabajo mediante plataformas de bajo código representa una oportunidad estratégica para optimizar procesos en instituciones de educación superior. En este contexto, se llevó a cabo la instalación, configuración y despliegue en producción de la plataforma n8n en el Instituto de Investigaciones Sociales de la Universidad Nacional Autónoma de México, como infraestructura de soporte para el “Taller Desarrollo de Aplicaciones con IA” de Educación Continua. La metodología siguió cuatro etapas: diseño de arquitectura mediante contenedores con Docker; configuración del entorno con variables seguras y *proxy* inverso Traefik; instalación verificada con certificados HTTPS automatizados mediante Let’s Encrypt; así como pruebas funcionales de integración con servicios externos. Se implementó adicionalmente un *script* de actualización semiautomática que respalda la base de datos PostgreSQL y los volúmenes de datos antes de aplicar nuevas versiones, lo cual permitió un mantenimiento controlado y reversible. Los resultados mostraron una plataforma estable con disponibilidad superior al 98% durante

el período de operación, capaz de ejecutar flujos de trabajo que integran modelos de lenguaje con servicios institucionales. Se concluye que el despliegue auto alojado de n8n constituye una alternativa viable y económica para instituciones académicas que requieren automatizar procesos de investigación e incorporar capacidades de inteligencia artificial en sus flujos de trabajo, al tiempo que se mantiene el control total sobre los datos institucionales.

Palabras clave: Automatización de bajo código, contenedores Docker, infraestructura como código, inteligencia artificial en educación, plataformas de bajo código.

Abstract

Automating workflows using low-code platforms presents a strategic opportunity for process optimization in higher education institutions. The installation, configuration and production deployment of the n8n platform was carried out at the Instituto de Investigaciones Sociales of the Universidad Nacional Autónoma de México, as the support infrastructure for the "Taller Desarrollo de Aplicaciones con IA" in Educación Continua. The implementation process consisted of four phases: designing a containerized architecture with Docker; configuring the environment with secure variables and a reverse proxy using Traefik; verifying the installation with automatic HTTPS certificates from Let's Encrypt; and testing functional integration with external services. In this context, a semi-automatic update script was also developed to back up the PostgreSQL data volume before applying new versions, thus facilitating reversible and controlled maintenance. The results demonstrated that the platform is stable, achieving over 98% availability during operation, and is capable of executing workflows that integrate language models with institutional services. It can be concluded that the self-deployment of n8n is a feasible and cost-effective alternative for academic institutions that require automation of research processes and seek to integrate artificial intelligence capabilities into their workflows, while maintaining full control over their own institutional data.

Keywords: Artificial intelligence in education, Docker containers, infrastructure as code, low-code platforms, low-code automation.

1. INTRODUCCIÓN

En los últimos años, se ha producido un creciente interés en las plataformas de automatización de flujos de trabajo de bajo código, tanto en el entorno empresarial como en el académico (Venkiteela, 2025; Joshi, 2025). La capacidad de integrar sistemas heterogéneos y servicios web mediante flujos visuales ha demostrado ser capaz de agilizar procesos repetitivos y reducir errores humanos (Hartz, 2025). De manera particular, en el ámbito de las operaciones de desarrollo, se ha encontrado que la configuración manual de *pipelines*, contenedores y despliegues incluye tareas repetitivas y propensas a error, por lo que la automatización resulta esencial para mejorar la fiabilidad y rapidez de entrega de *software* (Joshi, 2025). Diversos estudios recientes destacan la incorporación de plataformas nativas de la nube, como contenedores Docker y orquestadores tipo *Kubernetes*, junto con inteligencia artificial generativa en estas automatizaciones, lo que incrementa la eficiencia sin sacrificar seguridad ni gobernanza (Jennings, 2025; Traefik Labs, 2022).

La plataforma n8n se creó como una alternativa de código abierto de propósito general para orquestar flujos de trabajo complejos (Venkiteela, 2025). A diferencia de servicios propietarios de integración tipo

"If This Then That", n8n cuenta con una interfaz gráfica donde se integran y conectan fácilmente distintos servicios: APIs, bases de datos, aplicaciones en la nube e incluso lógica personalizada mediante código JavaScript. Esta plataforma permite el auto alojamiento con control total de los datos y una extensibilidad mediante nodos personalizados y API (Venkiteela, 2025). Según investigaciones especializadas, n8n es capaz de conectar aplicaciones empresariales con servicios de inteligencia artificial en un mismo flujo, lo que posibilita la creación de automatizaciones híbridas que combinan reglas determinísticas con razonamiento cognitivo. En pruebas comparativas, n8n ha demostrado un desempeño escalable y confiable, similar o superior al de otras herramientas similares como Node-RED, Apache Airflow o Zapier, especialmente debido a su modelo abierto y auto alojado (Venkiteela, 2025).

Es importante destacar que hay varias alternativas para implementar n8n. Por un lado, el servicio n8n Cloud permite trabajar de inmediato sin necesidad de realizar instalaciones, aunque impone algunas limitaciones según el plan contratado (como restricciones en la cantidad de ejecuciones y en la cantidad de flujos de trabajo activos). El plan Starter, por ejemplo, permite 2,500 ejecuciones mensuales y hasta cinco flujos de trabajo activos (n8n-io, 2024). Por otro lado, n8n puede alojarse en infraestructura propia o en un servicio de servidor virtual privado (VPS), lo que desbloquea flexibilidad, privacidad y escalabilidad. En este último caso, se tiene la libertad de ejecutar y activar cualquier número de flujos de trabajo, en donde el único límite es la potencia del equipo de cómputo elegido. El servicio de *hosting* VPS ofrece control y buen rendimiento, además de ser más asequible y escalable que un servidor dedicado; diversas compañías ofrecen estos servicios con diferentes planes que se adaptan a los requerimientos específicos de cada implementación.

Esta apertura y neutralidad por parte del proveedor hacen que n8n sea una plataforma atractiva en ambientes académicos, donde a menudo se requiere la integración de múltiples fuentes de datos y APIs de manera personalizada y con restricciones presupuestarias. De hecho, plataformas visuales como Node-RED ya han sido utilizadas en proyectos de investigación por su baja curva de aprendizaje y capacidad de combinar servicios diversos; por ejemplo, en bioinformática se ha utilizado Node-RED para flujos de análisis genómica, donde se obtuvo un resultado útil no sólo para especialistas en el campo sino también como herramienta educativa para estudiantes gracias a su interfaz visual intuitiva (Fàbregas, 2022). Estas características muestran que herramientas de flujo de trabajo de bajo código pueden trasladar la automatización a ambientes académicos, las cuales permiten a investigadores y personal técnico-académico implementar integraciones complejas sin necesidad de programación tradicional.

Por otro lado, la convergencia reciente entre la automatización de flujos de trabajo y la inteligencia artificial reafirma su relevancia en el ámbito académico. En ciencias sociales computacionales, se explora cómo los modelos de lenguaje avanzados pueden ser integrados en los flujos de trabajo para ayudar en tareas de análisis de datos, redacción de informes o formulación de hipótesis (Ziems *et al.*, 2024; Manning *et al.*, 2024). Se ha sugerido que los investigadores que aprovechen la automatización cognitiva, como agentes de IA integrados en plataformas de flujo, podrían aumentar significativamente su productividad (Korinek, 2023). Estas tendencias indican que la adopción de plataformas como n8n en el ámbito académico no sólo optimiza procesos técnicos, sino que también actúa como un puente formativo, ya que, al exponer al personal académico a herramientas de flujo visual con IA integrada, se facilita la comprensión práctica del funcionamiento de la inteligencia artificial y la mejora de procesos mediante automatización.

En el Instituto de Investigaciones Sociales de la Universidad Nacional Autónoma de México (IIS-UNAM) se identificó la necesidad de disponer de una plataforma de automatización que cumpliera con el doble propósito de ser infraestructura para proyectos de investigación y material didáctico para la capacitación

en tecnologías emergentes. En este sentido, el Departamento de Cómputo del IIS-UNAM desarrolló e impartió el “Taller Desarrollo de Aplicaciones con IA” a través de Educación Continua durante 2025, en el que se utilizó n8n como herramienta central para enseñar automatización de flujos de trabajo con integración de modelos de lenguaje. Sin embargo, la documentación disponible sobre la implementación de n8n en ambientes de producción académica se encontraba dispersa y enfocada fundamentalmente en entornos empresariales. Los asistentes al taller, investigadores y técnicos de ciencias sociales necesitaban una guía sistemática que abordara desde la implementación hasta el mantenimiento continuo, enfocada en seguridad y mejores prácticas para instituciones de educación superior. Por lo tanto, el objetivo fue implementar, configurar y desplegar en producción la plataforma n8n en el IIS-UNAM como infraestructura de automatización de flujos de trabajo con integración de inteligencia artificial, así como documentar un procedimiento sistemático y reproducible que otras instituciones académicas pudieran adaptar para implementar soluciones similares.

2. DESARROLLO TÉCNICO

2.1 METODOLOGÍA

Para la instalación y despliegue en producción de n8n, se siguió una metodología estructurada en cuatro etapas: diseño de la arquitectura, configuración del entorno, instalación de la solución y pruebas, junto con procedimientos de mantenimiento. El procedimiento se realizó sobre un servidor Linux (Ubuntu 22.04 LTS) configurado como servidor virtual privado con recursos de nivel producción (8 vCPU, 16 GB de RAM), a fin de asegurar el rendimiento y la estabilidad de la plataforma. Cabe mencionar que este procedimiento también es aplicable a equipos de cómputo local que cumplan con los requisitos mínimos de *hardware*: al menos, 4 vCPU y 8 GB de memoria RAM para una configuración básica, o bien, 8 vCPU y 16 GB de memoria RAM para ambientes de producción con mayor demanda.

Los requisitos adicionales considerados incluyeron: sistema operativo Ubuntu 22.04 o más reciente, un nombre de dominio registrado para apuntar al servicio (recomendado para certificados SSL), acceso SSH al servicio VPS, Docker y Docker Compose instalados y activos, así como familiaridad con comandos básicos de Linux y conocimiento de Docker. Estos prerrequisitos resultaron fundamentales para garantizar un despliegue exitoso y un mantenimiento adecuado de la plataforma a lo largo del tiempo.

2.1.1 DISEÑO

En la etapa de diseño se planeó la arquitectura del sistema, en la que se priorizó la seguridad, la persistencia de datos y la facilidad de mantenimiento. Se decidió utilizar un despliegue mediante contenedores utilizando Docker para envolver el n8n y sus dependencias, ya que la virtualización mediante contenedores permite tener entornos homogéneos y portátiles entre desarrollo y producción (Joshi, 2025; Jennings, 2025). Esta decisión se fundamentó en la necesidad de facilitar la creación, administración y ejecución del servicio de bases de datos y del servicio n8n en sí. Se definen tres componentes principales en un conjunto de contenedores Docker orquestados con Docker Compose: (1) un contenedor para la aplicación n8n; (2) un contenedor de base de datos PostgreSQL para almacenar de forma persistente la información de la plataforma —flujos de trabajo, credenciales cifradas e historiales—; y (3) un contenedor *proxy* inverso Traefik, encargado de gestionar el tráfico HTTP/HTTPS hacia n8n. Esta arquitectura modular asegura que

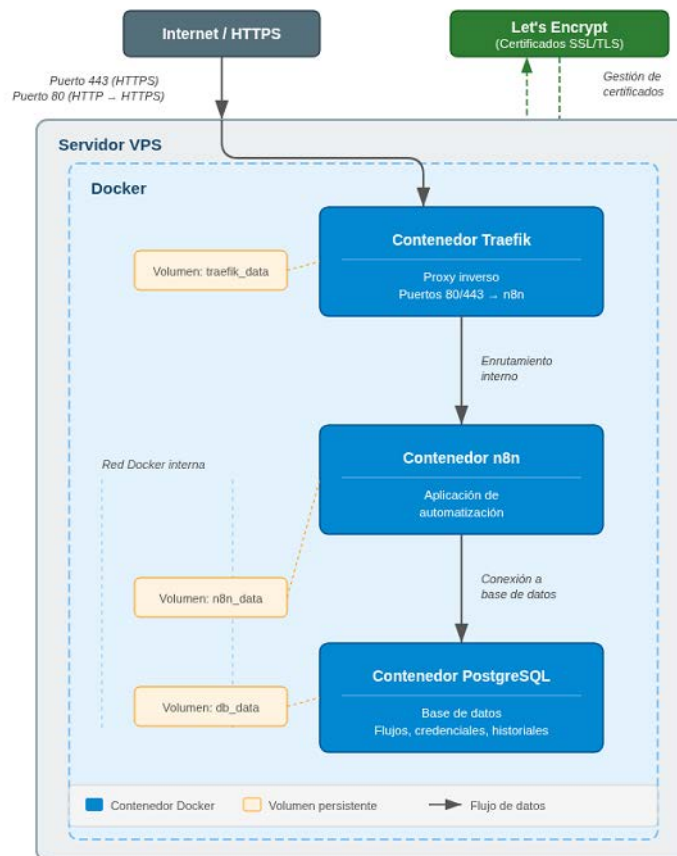
cada componente pueda actualizarse o reiniciarse de forma independiente y promueve la segregación de responsabilidades.

La elección de PostgreSQL como base de datos se basó en las recomendaciones oficiales de n8n para entornos productivos, donde una base SQL robusta ofrece mejor escalabilidad que el archivo SQLite predeterminado (n8n-io, 2024).

La arquitectura implementada se ilustra en la Figura 1, donde se observa la disposición de los contenedores Docker y el flujo de comunicación entre componentes. El servidor VPS aloja tres contenedores principales: Traefik como *proxy* inverso y gestor de certificados SSL/TLS, n8n como aplicación de automatización y PostgreSQL como sistema de gestión de base de datos. La red Docker interna permite la comunicación segura entre contenedores, mientras que Traefik gestiona el tráfico HTTPS externo y la renovación automática de certificados mediante Let's Encrypt.

Figura 1

Arquitectura de despliegue de n8n en producción mediante contenedores Docker



Se diseñó la persistencia de datos junto con el almacenamiento de la base de datos en un volumen Docker dedicado y un volumen adicional para los datos propios de n8n. Esto asegura que, en caso de reinicios o actualizaciones de contenedores, no se pierda la información de los flujos

ni las configuraciones del sistema. Esta consideración resultó especialmente importante dado el contexto académico, en el que la pérdida de flujos de trabajo desarrollados durante el taller hubiera tenido un impacto significativo en la experiencia de aprendizaje de los participantes. La decisión de utilizar contenedores Docker para el despliegue se fundamentó en las mejores prácticas documentadas para investigación reproducible, donde la virtualización mediante contenedores garantiza que el entorno de ejecución sea consistente y portable entre diferentes sistemas (Nüst *et al.*, 2020). Adicionalmente, se implementaron medidas de seguridad específicas para contenedores, considerando que, aunque la virtualización mediante contenedores proporciona aislamiento, el *software* malicioso puede explotar vulnerabilidades en imágenes de contenedores para atacar el sistema anfitrión (Devi Priya *et al.*, 2023)

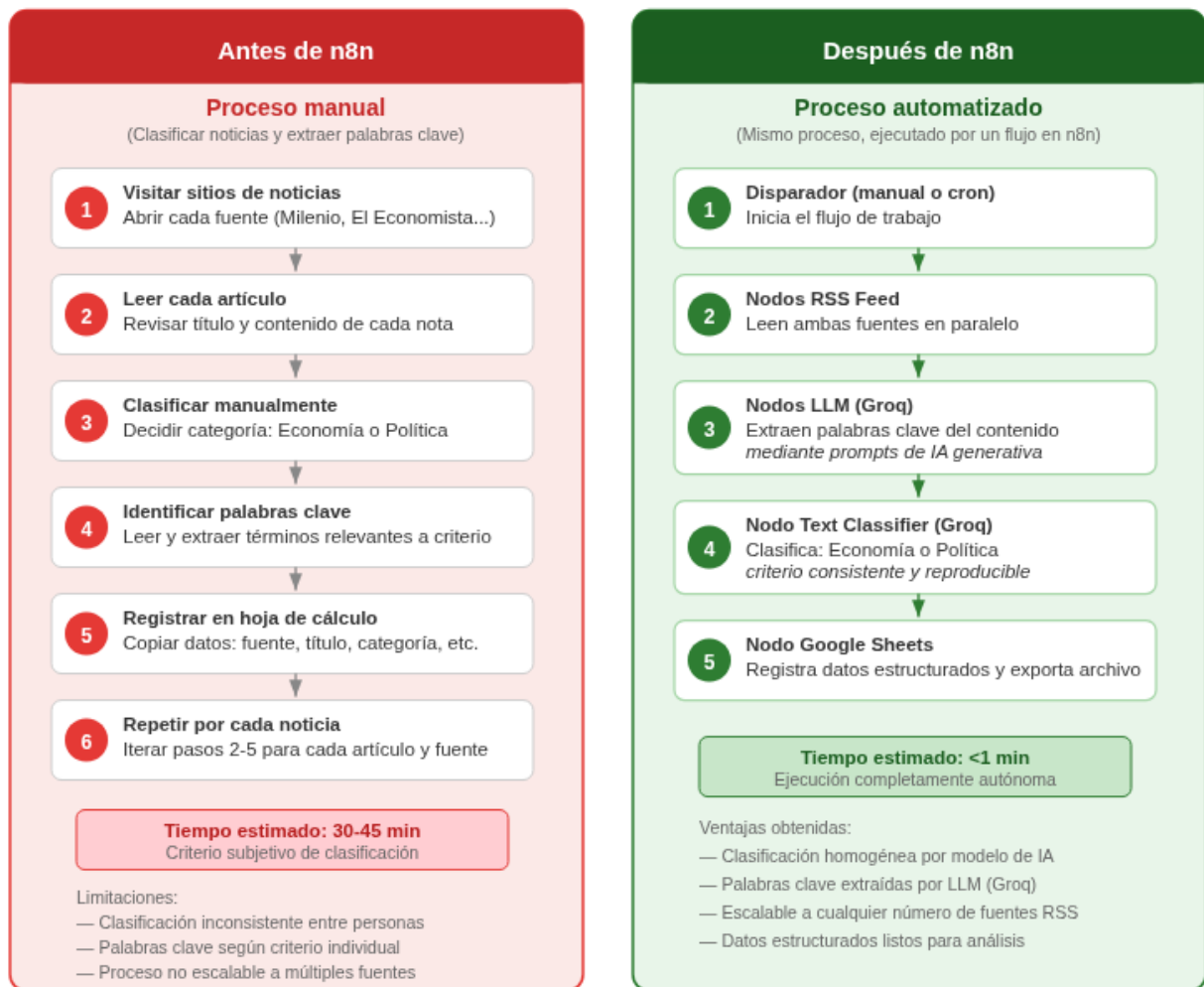
El *proxy* Traefik fue elegido en lugar de otros como Nginx por su integración nativa con Docker y soporte automático de HTTPS con Let's Encrypt (Traefik Labs, 2022). Se configuró Traefik para que monitoreara automáticamente los contenedores expuestos y gestionara los certificados TLS de manera automática, lo que facilitó la habilitación de la conexión segura, necesaria ya que algunas funcionalidades de n8n, como la autenticación OAuth2 con servicios de Google, requieren un *endpoint* HTTPS válido. Además, se aprovechó Traefik para centralizar la entrada a través de los puertos 80/443 del servidor, manteniendo al contenedor de n8n sin exposición directa a Internet, lo que permitió añadir una capa de seguridad adicional al reducir la superficie de ataque potencial.

En lo que respecta a la seguridad y control de acceso, se planificaron varias medidas desde la etapa de diseño. En primer lugar, se contempló la habilitación de la autenticación básica HTTP en n8n para proteger la interfaz web con credenciales durante el primer arranque del sistema. Cabe señalar que, en versiones recientes de n8n, estas variables de autenticación son reconocidas únicamente durante la configuración inicial, ya que, una vez creada la cuenta de propietario a través de la interfaz web, el sistema nativo de gestión de usuarios (*User Management*), basado en base de datos, toma precedencia y la autenticación básica deja de utilizarse. Asimismo, se diseñó la utilización de variables de entorno para gestionar la información sensible (contraseñas de base de datos, claves de API, usuario/contraseña de autenticación), manteniéndolas fuera de los archivos de configuración en texto plano. Finalmente, se diseñó un esquema de actualización y mantenimiento mediante un *script* de actualización semiautomático capaz de respaldar datos y renovar la imagen de n8n. De esta manera, se aseguró que la plataforma pudiera mantenerse actualizada con un mínimo de intervención y riesgo.

La Figura 2 ilustra la comparación entre el proceso manual y el proceso automatizado para una tarea representativa del taller: la clasificación de noticias y la extracción de palabras clave a partir de fuentes RSS. En el esquema manual, cada paso requería intervención humana directa, desde la visita a los sitios de noticias hasta el registro en hoja de cálculo, con un tiempo estimado de 30 a 45 minutos y criterios de clasificación subjetivos que variaban entre personas. Con la implementación de n8n, el mismo proceso se ejecutó de manera autónoma mediante un flujo de trabajo que integra nodos de lectura de fuentes RSS, extracción de palabras clave a través de un modelo de lenguaje (Groq) y clasificación automática de las noticias en categorías (Economía o Política) mediante un nodo de clasificación de texto con IA. Los resultados fueron consistentes y reproducibles, registrados automáticamente en Google Sheets.

Figura 2

Comparación del proceso de clasificación de noticias y extracción de palabras clave antes y después de la implementación de n8n



Este contraste evidencia el potencial de la plataforma para transformar tareas manuales repetitivas en procesos automatizados con criterios homogéneos.

2.1.2 CONFIGURACIÓN

Después de la planificación, se procedió a establecer el entorno de acuerdo con lo diseñado. En primer lugar, se configuró el servidor VPS con los prerequisites básicos. Para instalaciones en equipos de cómputo locales, se ejecutaron los siguientes comandos de instalación de Docker Engine, containerd y Docker Compose, en donde se agregó primero el repositorio que permite la descarga del *software*; en el Anexo A del reporte, se presentan detalles adicionales sobre este tema.

Para servicios VPS, Docker generalmente viene preinstalado, por lo que únicamente se verificó que estuviera activo mediante el comando `docker --version` —si fuera necesario, se inicia el servicio con `sudo systemctl start docker`—. Después, se configuró un nombre de dominio (subdominio institucional), el cual apunta a la dirección IP del servidor, dado que la emisión de certificados TLS de Let's Encrypt requiere validar el dominio.

Con el dominio listo, se creó el archivo de entorno (.env) para almacenar las variables de configuración sensibles y específicas del despliegue. Entre ellas, se incluyeron: el subdominio (como parte de .unam.mx), las credenciales de base de datos (usuario, contraseña y nombre de base de datos PostgreSQL), la zona horaria, así como las credenciales para la autenticación básica de n8n, las cuales aplican únicamente durante el primer inicio del sistema antes de la creación de la cuenta de propietario. En este archivo, se incluyeron variables propias de n8n como la activación del modo de autenticación básica, el protocolo HTTPS y la URL base para *webhooks*. Estas configuraciones garantizaron que n8n reconociera su entorno de despliegue en producción y generara enlaces correctos para funciones especiales como URLs de *webhook* o autenticaciones externas.

Por último, se creó el directorio de proyecto y el archivo `docker-compose.yml` con la configuración de los tres servicios. Primero, se creó la estructura de directorios (en el Anexo B del reporte se detalla este punto), y luego se procedió con la configuración del archivo `docker-compose.yml`, en donde se incluyeron los tres servicios principales mencionados (en el Anexo C del reporte se presentan detalles adicionales acerca de este tema).

2.1.3 INSTALACIÓN

Una vez finalizado el entorno, se pasó a la instalación y despliegue propiamente dicho. Todos los servicios se iniciaron mediante Docker Compose con el siguiente comando:

```
docker compose up -d
```

Docker Compose descargó automáticamente las imágenes de n8n, PostgreSQL y Traefik desde el registro oficial de Docker Hub y se procedió a crear las instancias de los contenedores según la configuración definida. Gracias a la directiva *healthcheck*, el inicio fue secuencial y controlado: primero, PostgreSQL se puso en estado saludable después de comprobar repetidamente que el servicio estuviera listo para aceptar conexiones, lo que permitió al servicio n8n conectarse con la base de datos en funcionamiento sin problemas de conexión.

Por su parte, Traefik comprobó, mediante las etiquetas definidas en el servicio n8n, que debía enrutar el subdominio configurado al contenedor y, en ese momento, generó automáticamente una solicitud a Let's Encrypt para obtener el certificado SSL del dominio. Este proceso de emisión de certificado se completó satisfactoriamente en cuestión de segundos. Dichos certificados fueron almacenados en el volumen correspondiente, lo que permitió su reutilización y renovación automática antes del vencimiento, típicamente cada 90 días según las políticas de Let's Encrypt.

Una vez iniciados los tres contenedores, se comprobó el estado mediante comandos de inspección como `docker compose ps` y `docker compose logs`. Las bitácoras confirmaron que n8n se inició en modo producción, escuchando en el puerto interno configurado, y que Traefik expuso correctamente el enrutador con certificado válido. No se encontraron problemas en el inicio, mientras que PostgreSQL

confirmó la creación de la base de datos indicada y la aceptación de conexiones desde el contenedor de n8n.

2.1.4 PRUEBAS

Después de la instalación, se realizaron pruebas de funcionamiento y seguridad para comprobar que el despliegue se ajustaba a los requisitos establecidos. En primer lugar, se accedió a través de un navegador web al dominio configurado para comprobar la carga de la interfaz de n8n. El navegador reflejó el certificado TLS emitido por Let's Encrypt como válido, lo que indicaba que la comunicación se realizaba cifrada de forma correcta a través de HTTPS. Esta comprobación fue especialmente relevante dado que varios participantes en el taller accedieron a la plataforma desde redes externas a la universidad.

Al acceder por primera vez, n8n mostró una secuencia de formularios de configuración inicial que guiaron al administrador en el proceso de inicio. El primer formulario solicitó la creación de la cuenta de propietario, donde se requería dirección de correo electrónico, nombre y apellido, así como contraseña segura con un mínimo de ocho caracteres y, al menos, un número y una letra mayúscula. Este paso era necesario para crear al usuario administrador de la instancia.

El segundo formulario era una secuencia de preguntas de personalización de la instancia, diseñadas para adaptar la experiencia de usuario: tipo de organización, rol del usuario, finalidad principal de las automatizaciones, tamaño de la organización y cómo se conoció n8n. Aunque estas preguntas no eran necesarias para la funcionalidad, facilitaron la personalización de la interfaz.

El tercer paso proporcionó la posibilidad de desbloquear características gratuitas adicionales a través del registro de la instancia, como historial de ejecución extendido, herramientas de depuración avanzadas y estructura de carpetas para la organización de flujos de trabajo. Se introdujo una cuenta de correo institucional para obtener una licencia gratuita. Posteriormente, se verificó la bandeja de entrada del correo proporcionado para obtener el código de activación, denominado *Activation Key*, que se ingresó en la ruta *Settings* → *Usage and Plan* → *Enter Activation Key*; de esta manera, la instancia pasó a ser considerada registrada, la cual desbloqueó características avanzadas por tiempo ilimitado.

Una vez verificada la accesibilidad de la interfaz web, se procedió a probar la ejecución de flujos de trabajo básicos. Se creó un flujo sencillo con nodos de ejemplo —un disparador manual y un nodo que retornaba un mensaje de texto— para probar que el motor de ejecución funcionaba correctamente. El flujo se ejecutó sin problemas y los registros de n8n mostraron la secuencia de pasos completados con sus tiempos de ejecución.

También, se probó la integración con servicios externos a través de un nodo de solicitud HTTP que apunta a una API pública de prueba. La ejecución obtuvo la respuesta esperada, lo que indicó que n8n podía realizar solicitudes salientes a Internet sin restricciones del *firewall* o la configuración de red. Además, se probó una autenticación OAuth2 contra los servicios de Google para corroborar que n8n podía manejar el ciclo completo de autorización; la redirección de Google reconoció el dominio HTTPS como válido y permitió completar la conexión exitosamente.

Finalmente, se llevó a cabo una prueba de resistencia y estabilidad a largo plazo: se dejó correr el contenedor de n8n durante varias semanas con flujos programados mediante disparadores de tipo cron, incluyendo flujos que enviaban recordatorios por correo y flujos que consultaban APIs externas periódicamente. Se monitoreó el consumo de recursos mediante herramientas como *docker stats* y se

verificó la estabilidad del sistema. Los resultados fueron satisfactorios: n8n mantuvo un uso moderado de CPU y memoria acorde a lo esperado (menos del 10% de CPU en reposo y picos controlados durante ejecuciones); no se observaron caídas del servicio y la base de datos PostgreSQL mostró estabilidad en la inserción y consulta periódica de datos sin degradación del rendimiento.

2.2 MANTENIMIENTO Y ACTUALIZACIÓN

Como parte integral de la metodología, se implementó un procedimiento de mantenimiento continuo con énfasis en la actualización segura de la plataforma n8n a futuras versiones. El equipo de n8n, junto con la comunidad de desarrollo, publica actualizaciones frecuentes que incluyen correcciones de seguridad, nuevas funcionalidades e integraciones con servicios externos y mejoras de rendimiento. Debido a esto, mantener la plataforma actualizada resulta importante para aprovechar estas mejoras y proteger la infraestructura.

Con el fin de automatizar gran parte del proceso de actualización, tanto en ambiente local de pruebas como en producción, se desarrolló un *script* Bash denominado *update_core.sh* con ayuda de IA generativa. Este *script* fue diseñado con un enfoque preventivo que prioriza la preservación de datos sobre la velocidad de actualización.

Entre las funcionalidades implementadas en el *script*, se encuentran las siguientes: (1) respaldo de base de datos, en donde se utiliza el volcado de PostgreSQL con el comando *pg_dump* ejecutado dentro del contenedor que almacena el respaldo SQL compactado con fecha en un directorio específico para respaldos; (2) respaldo de volúmenes, en donde se almacena el contenido del volumen de datos de n8n en un archivo compactado; (3) detección automática de servicios definidos en el archivo *docker-compose.yml*, donde se incluyen servicios opcionales tales como bases de datos vectoriales (Qdrant) o servidores de modelos locales (Ollama) que podrían agregarse en un futuro; (4) actualización de imágenes Docker, sólo después de que se hayan realizado de manera exitosa los respaldos, donde se obtienen las nuevas versiones con comandos de Docker Compose; (5) verificación de salud post-actualización, donde se conecta internamente a los *endpoints* de cada servicio hasta obtener una respuesta satisfactoria, con reintentos y tiempos de espera configurables; y (6) registro detallado de operaciones en un archivo de log con sello de tiempo, que permite a los administradores revisar cualquier incidencia ocurrida durante la actualización.

Es importante destacar que, aunque el *script* automatiza la secuencia de respaldo, descarga y actualización, la toma de decisión para llevar a cabo la actualización es manual e informada. A través de las mejores prácticas de DevOps, documentadas en la literatura (Saini, 2025), primero se probó la nueva versión de n8n en un entorno de pruebas local o en una instancia secundaria para asegurarse de su estabilidad y compatibilidad con los flujos existentes. Sólo después de validar en este entorno de laboratorio que la versión funcionaba correctamente con los nodos y credenciales utilizados en producción, se procedió a ejecutar el *script* en el servidor de producción.

Esta precaución se originó después de que una experiencia durante el período de funcionamiento demostrara su necesidad: una versión particular de n8n presentó un error inesperado en la interfaz de credenciales, impidiendo llenar algunos campos necesarios para conectarse con APIs externas (Cristiangrando, 2025). Este *bug* afectó específicamente a la creación de nuevas credenciales, debido a que las existentes funcionaban con normalidad. Ese problema hubiera afectado gravemente a los flujos de trabajo del taller —donde los participantes necesitaban configurar nuevas conexiones

con servicios de Google Gemini y OpenRouter—, pero la política de prueba previa evitó una interrupción significativa. A raíz de esta experiencia, se consolidó la política de “actualización semiautomática supervisada”: el *script* asegura rapidez y consistencia en la actualización (junto con la posibilidad de reversión mediante los respaldos en caso de ser necesario), pero la activación de la actualización requiere ejecución humana tras pruebas satisfactorias en un entorno controlado. La adopción de estas prácticas DevOps responde a la creciente importancia de esta metodología, tanto en la industria como en contextos académicos donde se ha documentado que las instituciones de educación superior están incorporando gradualmente principios DevOps en sus infraestructuras tecnológicas. Persisten, sin embargo, brechas tecnológicas entre la formación académica y las demandas de la industria (Sánchez-Cifo *et al.*, 2023).

3. RESULTADOS

El despliegue de n8n en el servidor del IIS-UNAM se llevó a cabo de manera exitosa y la plataforma se puso en marcha como infraestructura de soporte para el “Taller Desarrollo de Aplicaciones con IA” de Educación Continua durante 2025. A continuación, se presentan los resultados obtenidos durante el período de funcionamiento.

3.1 DISPONIBILIDAD Y ESTABILIDAD

Durante el período de funcionamiento (enero a diciembre de 2025), la plataforma logró una disponibilidad superior al 98%. Las interrupciones registradas fueron exclusivas a actualizaciones programadas del sistema operativo y de la plataforma n8n en sí misma, así como a un reinicio del servidor por mantenimiento de seguridad programado por la administración de la infraestructura universitaria. El *script* de actualización fue ejecutado en varias ocasiones a lo largo del año para aplicar actualizaciones de seguridad y funcionalidades nuevas de n8n, sin pérdida de datos ni flujos de trabajo a través del sistema de respaldos implementado. En todos los casos, el tiempo de inactividad no superó los 15 minutos por evento de mantenimiento.

3.2 USO EN EL TALLER DE CAPACITACIÓN

La plataforma se utilizó como espacio de práctica para las 16 sesiones del taller, en los cuales los participantes (investigadores y técnicos académicos del área de ciencias sociales) aprendieron a configurar credenciales de API para servicios de inteligencia artificial (Google Gemini, OpenRouter), a diseñar flujos de trabajo que integren modelos de lenguaje con hojas de cálculo de Google, a implementar memoria en agentes conversacionales para mantener contexto entre interacciones y a automatizar tareas repetitivas de recolección y procesamiento de datos relevantes para sus proyectos de investigación.

Los participantes lograron diseñar y ejecutar flujos de trabajo funcionales sin necesitar conocimientos previos de programación, lo que validó la hipótesis de que las plataformas de bajo código pueden democratizar la automatización en ambientes académicos (Venkiteela, 2025). El taller se dividió en tres módulos temáticos complementarios: el primer módulo, enfocado en la automatización con n8n, se dividió en ocho sesiones, en las que los participantes aprendieron los conceptos básicos de flujos de trabajo (nodos, conexiones, disparadores y acciones) y se trató ampliamente con el concepto de JSON como formato de intercambio de datos fundamental para la comunicación entre servicios; el segundo

módulo se enfocó en herramientas de prototipado rápido que complementaron las capacidades de automatización; y el tercer módulo se enfocó en el uso de entornos de desarrollo asistidos por IA para casos que requirieron personalización más allá de las capacidades de bajo código.

Un hallazgo relevante fue la curva de aprendizaje diferenciada observada entre los participantes: aquellos sin conocimientos previos en programación lograron diseñar flujos funcionales a lo largo de las sesiones mediante la interfaz visual, mientras que los participantes con conocimientos básicos en programación pudieron diseñar automatizaciones más complejas en donde se aprovechó los nodos de código JavaScript disponibles en n8n. Este hallazgo sugiere que, efectivamente, n8n reduce la barrera de entrada para la automatización de procesos, aunque no la elimina completamente —los participantes aún tuvieron que comprender conceptos fundamentales como APIs, métodos de autenticación (claves de API vs. OAuth2) y estructuras de datos en formato JSON—. Este patrón de aprendizaje es consistente con investigaciones recientes sobre el desarrollo de habilidades metacognitivas en plataformas de bajo código en contextos educativos, donde se ha documentado que la interfaz visual facilita la comprensión de conceptos abstractos de programación sin eliminar la necesidad de comprender la lógica subyacente (Matook *et al.*, 2024).

3.3 INTEGRACIÓN CON SERVICIOS DE IA

Se verificó la correcta integración de n8n con los siguientes servicios de inteligencia artificial y productividad, todos ellos utilizados activamente durante las sesiones del taller (el Anexo A del reporte presenta detalles adicionales acerca de este punto).

La implementación de HTTPS probó ser fundamental para las integraciones OAuth2, lo que ratificó la estrategia de diseño de implementar Traefik con certificados SSL automáticos desde el comienzo del proyecto. De no tener el certificado SSL válido, las integraciones con servicios de Google no habrían sido posibles, lo que habría limitado severamente la utilidad de la plataforma para el taller.

3.4 COMPARACIÓN CON ALTERNATIVAS

El auto alojamiento de n8n probó tener ventajas sustanciales frente a las alternativas evaluadas inicialmente, las cuales se resumen en la tabla comparativa que se encuentra en el Anexo E.

Para el contexto académico del IIS-UNAM, el modelo auto alojado en servidor local resultó el más adecuado tanto por consideraciones presupuestarias —dado que el costo del VPS es significativamente menor que las suscripciones a servicios en la nube para el volumen de ejecuciones requerido— como por el requisito de mantener control sobre datos de investigación potencialmente sensibles, lo cual es consistente con las recomendaciones de la literatura sobre autonomía de datos en entornos académicos (Venkiteela, 2025).

3.5 ANÁLISIS DE RESULTADOS

Los resultados obtenidos confirman lo reportado en la literatura respecto a la viabilidad de n8n en entornos productivos con cargas de trabajo moderadas a intensas. La tasa de disponibilidad superior al 98% es consistente con los *benchmarks* publicados para despliegues similares en infraestructura de nivel producción. Asimismo, la experiencia del taller demostró empíricamente que investigadores sin formación técnica formal en programación pueden diseñar y ejecutar automatizaciones complejas en períodos de

capacitación relativamente cortos (16 sesiones de dos horas cada una), en relación con los hallazgos teóricos sobre el potencial de las plataformas de bajo código para democratizar la automatización y la integración de inteligencia artificial en flujos de trabajo de investigación (Venkiteela, 2025).

El caso del *bug* en la versión de *n8n* que afectaba la interfaz de credenciales (Cristiangrando, 2025) ratificó posteriormente la política de actualización supervisada que se implementó desde el inicio del proyecto. Si se hubiera actualizado de manera automática sin comprobar previamente en un entorno de pruebas, los asistentes al taller no hubieran podido establecer nuevas conexiones con APIs externas durante varias sesiones, lo que hubiera tenido un impacto significativo en la experiencia de aprendizaje y el cumplimiento de los objetivos pedagógicos del taller.

4. CONCLUSIONES

Con el uso del *n8n* en el Instituto de Investigaciones Sociales de la UNAM se pudo proveer una herramienta de automatización de flujos de trabajo con inteligencia artificial integrada para uso académico y formativo. Los resultados obtenidos durante el tiempo de funcionamiento demostraron la factibilidad técnica y pedagógica del enfoque utilizado.

La arquitectura, contenerizada con Docker, PostgreSQL como base de datos y Traefik como *proxy* inverso, resultó ser una combinación sólida que facilitó tanto el proceso de implementación inicial como el mantenimiento continuo. La implementación de certificados HTTPS automatizada con Let's Encrypt eliminó un obstáculo técnico significativo para la integración con servicios externos que requieren conexiones seguras, especialmente aquellos que utilizan OAuth2 como mecanismo de autenticación.

El *script* de actualización semiautomática, desarrollado con asistencia de IA generativa, es una herramienta útil para administradores de sistemas que manejan instancias de *n8n* en producción. El enfoque de respaldar antes de actualizar con verificación manual previa en ambiente de pruebas ofrece un equilibrio apropiado entre mantenerse actualizado con las últimas actualizaciones de correcciones de seguridad y las nuevas funcionalidades, al tiempo que se minimizan los riesgos de interrupción del servicio por incompatibilidades o errores introducidos en nuevas versiones.

La experiencia del taller ratificó que las plataformas de automatización de bajo código pueden funcionar como puentes formativos para acercar a investigadores de ciencias sociales a las tecnologías de inteligencia artificial que los potencia para diseñar soluciones a problemas específicos de su disciplina sin requerir conocimientos avanzados de programación. Esta democratización de la automatización tiene implicaciones importantes para la productividad de la investigación académica en el contexto actual de transformación digital.

Se recomienda a otras instituciones de educación superior interesadas en implementar soluciones similares considerar los siguientes aspectos derivados de la experiencia documentada: (a) planificar la arquitectura con énfasis en persistencia de datos y seguridad desde el diseño inicial, junto con el uso de volúmenes Docker dedicados y variables de entorno para información sensible; (b) implementar procedimientos de respaldo automatizado antes de cualquier actualización, se incluye tanto la base de datos como los volúmenes de configuración; (c) mantener un entorno de pruebas paralelo para validar nuevas versiones antes de aplicarlas en producción, especialmente para actualizaciones mayores; y (d)

documentar los procedimientos de manera que sean replicables por personal técnico con conocimientos básicos de administración de sistemas Linux y Docker.

Durante las pruebas, se identificó una limitación importante: el servidor VPS donde se alojó n8n no contaba con recursos suficientes para ejecutar modelos de lenguaje locales mediante Ollama. Incluso modelos relativamente pequeños, de 8 mil millones de parámetros, presentaron un rendimiento insuficiente para atender consultas simples o conversaciones básicas, lo cual descartó inicialmente la posibilidad de independizarse de APIs externas para ciertos casos de uso.

Como alternativa, se efectuaron pruebas exitosas con LM Studio en una Mac mini M4 Pro con 24 GB de memoria RAM unificada, se logró un desempeño aceptable de 68 *tokens* por segundo y 3.77 segundos para el primer *token* en consultas simples. Los proyectos futuros incluirán la implementación de una arquitectura que permitirá a n8n acceder a modelos locales ejecutados en este equipo como si fueran un servicio interno. La arquitectura permitirá mantener el principio de control institucional sobre los datos y las automatizaciones, lo que permitirá ejecutar modelos de lenguaje sin depender de servicios externos ni exponer la infraestructura de inferencia a Internet.

Desde una perspectiva más amplia, la experiencia documentada en este reporte contribuye al creciente cuerpo de conocimiento sobre la adopción de tecnologías de inteligencia artificial en instituciones de educación superior mexicanas. La combinación de infraestructura auto alojada con capacitación práctica representa un modelo que otras áreas universitarias podrían replicar y adaptar a sus necesidades específicas de investigación y docencia. La documentación detallada de los procedimientos técnicos, junto con el *script* de mantenimiento, busca facilitar esta transferencia de conocimiento y reducir la curva de aprendizaje para equipos técnicos que deseen implementar soluciones similares en sus propios contextos institucionales.

AGRADECIMIENTOS

Se agradece al Instituto de Investigaciones Sociales de la UNAM por el apoyo institucional para el desarrollo del “Taller Desarrollo de Aplicaciones con IA” a través de Educación Continua, y a la Mtra. Sofía Aké Farfán, jefa del Departamento de Cómputo, por las facilidades otorgadas en infraestructura de cómputo. Asimismo, se reconoce la participación entusiasta de los asistentes al taller, cuyas preguntas y casos de uso enriquecieron la implementación documentada en este reporte.

Declaración de contribución de autoría

Ernesto Pathros Ibarra García: Docente principal del taller; diseñador del currículo académico; especialista técnico en integración de servicios de IA. Realizó la automatización de procesos con inteligencia artificial; desarrollo de aplicaciones inteligentes; educación en tecnologías de información y comunicación. Lideró el diseño del módulo de automatización del taller “Desarrollo de Aplicaciones con IA” y definió los objetivos de aprendizaje y la secuencia didáctica de las 16 sesiones impartidas. Desarrolló con IA generativa el *script* de actualización semiautomática *update_core.sh* documentado en este reporte, así como la implementación la lógica de respaldo de base de datos, respaldo de volúmenes y verificación de salud post-actualización. Contribuyó a las secciones del artículo sobre configuración de APIs de inteligencia artificial (Google Gemini, OpenRouter), mejores prácticas de seguridad en la plataforma n8n y análisis de la experiencia docente que motivó el reporte técnico.

Julio César Cruz Estrada: Docente del taller; especialista en infraestructura y despliegue; responsable técnico de instalación del servidor. Aportó en la infraestructura de tecnologías de información y comunicación; DevOps; administración de sistemas; despliegue en producción. Lideró la implementación de n8n en la infraestructura del servidor del IIS-UNAM y la ejecución del aprovisionamiento del VPS, la configuración de Docker y Docker Compose, así como el despliegue inicial de los contenedores. Elaboró la documentación detallada de los procedimientos de instalación tanto en modo local como en servidor de producción, que sirvió como base para este reporte técnico. Participó como docente en las sesiones del taller enfocadas a configuración de credenciales y resolución de problemas técnicos identificados por los participantes durante los ejercicios prácticos.

REFERENCIAS

- Cristiangrando. (2025). *Visual bug when registering credentials or auth* (Issue #20279). GitHub. <https://github.com/n8n-io/n8n/issues/20279>
- Devi Priya, V. S., Sethuraman, S. C., & Khan, M. K. (2023). Container security: Precaution levels, mitigation strategies, and research perspectives. *Computers & Security*, 135, 103490. <https://doi.org/10.1016/j.cose.2023.103490>
- Fàbregas, F. (2022). *Development of Node-RED web-based, open-source visual workflow tool for bioinformatics* [Trabajo de fin de máster, Universitat Oberta de Catalunya]. Repositorio Institucional UOC. <https://openaccess.uoc.edu/handle/10609/146604>
- Hartz, E. L. (2025). *Development of an AI-driven DevOps Engineer: Automating workflows with an LLM based multi-agent system* [Tesis de maestría, Universidad Técnica de Múnich]. DiVA Portal. <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1987584>
- Jennings, T. (2025). *Building strong container security for modern applications*. Mend.io. <https://www.mend.io/blog/building-strong-container-security/>
- Joshi, S. (2025). A review of generative AI and DevOps pipelines: CI/CD, agentic automation, MLOps integration, and LLMs. *International Journal of Innovative Research in Computer Science and Technology*, 13(4), 1-14. <https://doi.org/10.55524/ijrcst.2025.13.4.1>
- Korinek, A. (2023). Generative AI for economic research: Use cases and implications for economists. *Journal of Economic Literature*, 61(4), 1281-1317. <https://doi.org/10.1257/jel.20231736>
- Manning, B. S., Zhu, K., & Horton, J. J. (2024). *Automated social science: Language models as scientist and subjects* [Documento de trabajo]. National Bureau of Economic Research. <https://www.nber.org/papers/w32381>
- Matook, S., Maggie Wang, Y., Koeppel, N., & Guerin, S. (2024). Metacognitive skills in low-code app development: Work-integrated learning in information systems development. *Journal of Information Technology*, 39(1), 41-70. <https://doi.org/10.1177/02683962231170238>
- n8n-io. (2024). *Self-hosted AI starter kit*. GitHub. <https://github.com/n8n-io/self-hosted-ai-starter-kit>
- Nüst, D., Sochat, V., Marwick, B., Eglén, S. J., Head, T., Hirst, T., & Evans, B. D. (2020). Ten simple rules for writing Dockerfiles for reproducible data science. *PLOS Computational Biology*, 16(11), e1008316.

<https://doi.org/10.1371/journal.pcbi.1008316>

- Saini, A. (2025). Continuous Integration & Continuous Deployment (CI/CD) best practices to follow in 2026. *Kellton Tech Blog*. <https://www.kellton.com/kellton-tech-blog/continuous-integration-deployment-best-practices-2025>
- Sánchez-Cifo, M. Á., Bermejo, P., & Navarro, E. (2023). DevOps: Is there a gap between education and industry? *Journal of Software: Evolution and Process*, 35(12), e2534. <https://doi.org/10.1002/smr.2534>
- Traefik Labs. (2022). *HTTPS encryption: TLS, SSL, and Let's Encrypt*. Traefik Labs Glossary. <https://traefik.io/glossary/https-encryption-tls-ssl-letsencrypt>
- Venkiteela, P. (2025). n8n: An open-source workflow automation platform for enterprise integration and AI-driven orchestration. *International Journal of Computer Applications*, 187(63). <https://doi.org/10.5120/ijca2025926031>
- Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., & Yang, D. (2024). Can large language models transform computational social science? *Computational Linguistics*, 50(1), 237-291. https://doi.org/10.1162/coli_a_00502

ANEXO A. LISTADO DE COMANDOS BÁSICOS PARA LA INSTALACIÓN DE DOCKER EN EL SERVIDOR

Agregar la llave GPG oficial de Docker:

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg \  
-o /etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Agregar el repositorio a las fuentes de Apt:

```
echo "deb [arch=$(dpkg --print-architecture) \  
signed-by=/etc/apt/keyrings/docker.asc] \  
https://download.docker.com/linux/ubuntu \  
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get update
```

```
sudo apt-get update
```

Instalar Docker:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io \  
docker-buildx-plugin docker compose-plugin
```

```
docker-buildx-plugin docker compose-plugin
```

Iniciar y habilitar el servicio:

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

Agregar usuario al grupo docker:

```
sudo groupadd docker
```

```
sudo usermod -aG docker usuario_admin
```

```
newgrp docker
```

ANEXO B. LISTADO DE COMANDOS PARA LA CREACIÓN DE DIRECTORIOS

```
mkdir ~/n8n && cd ~/n8n  
touch docker-compose.yml
```

ANEXO C. CONFIGURACIÓN DEL ARCHIVO DOCKER-COMPOSE.YML

```
version: '3.8'  
services:  
  traefik:  
    image: traefik:v2.9  
    restart: unless-stopped  
    command:  
      --api=true  
      --api.insecure=true  
      --providers.docker=true  
      --providers.docker.exposedbydefault=false  
      --entrypoints.websecure.address=:443  
      --certificatesresolvers.mytlschallenge.acme.tlschallenge=true  
      --certificatesresolvers.mytlschallenge.acme.email=${EMAIL}  
      --certificatesresolvers.mytlschallenge.acme.storage=/letsencrypt/acme.json  
    ports:  
      - 443:443  
      - 80:80  
    volumes:  
      - traefik_data:/letsencrypt  
      - /var/run/docker.sock:/var/run/docker.sock:ro  
n8n:  
  image: n8nio/n8n:latest
```

restart: unless-stopped

environment:

- GENERIC_TIMEZONE=\${GENERIC_TIMEZONE}
- TZ=\${GENERIC_TIMEZONE}
- N8N_HOST=\${SUBDOMAIN}.\${DOMAIN_NAME}
- N8N_PORT=443
- N8N_PROTOCOL=https
- NODE_ENV=production
- WEBHOOK_URL=https://\${SUBDOMAIN}.\${DOMAIN_NAME}/
- N8N_BASIC_AUTH_ACTIVE=\${N8N_BASIC_AUTH_ACTIVE}
- N8N_BASIC_AUTH_USER=\${N8N_BASIC_AUTH_USER}
- N8N_BASIC_AUTH_PASSWORD=\${N8N_BASIC_AUTH_PASSWORD}
- N8N_DATA_TABLES_MAX_SIZE_BYTES=\${N8N_DATA_TABLES_MAX_SIZE_BYTES}
- DB_TYPE=\${DB_TYPE}
- DB_POSTGRESDB_DATABASE=\${DB_POSTGRESDB_DATABASE}
- DB_POSTGRESDB_HOST=\${DB_POSTGRESDB_HOST}
- DB_POSTGRESDB_PORT=\${DB_POSTGRESDB_PORT}
- DB_POSTGRESDB_USER=\${DB_POSTGRESDB_USER}
- DB_POSTGRESDB_PASSWORD=\${DB_POSTGRESDB_PASSWORD}

volumes:

- n8n_data:/home/node/n8n
- ./local-files:/files

labels:

- traefik.enable=true
- traefik.http.routers.n8n.rule=Host(`\${SUBDOMAIN}.\${DOMAIN_NAME}`)
- traefik.http.routers.n8n.tls=true
- traefik.http.routers.n8n.entrypoints=websecure
- traefik.http.routers.n8n.tls.certresolver=mytlschallenge

depends_on:

- db

- traefik

db:

image: postgres:14

restart: unless-stopped

environment:

- POSTGRES_USER=\${DB_POSTGRESDB_USER}

- POSTGRES_PASSWORD=\${DB_POSTGRESDB_PASSWORD}

- POSTGRES_DB=\${DB_POSTGRESDB_DATABASE}

volumes:

- db_data:/var/lib/postgresql/data

healthcheck:

test: ["CMD-SHELL", "pg_isready -U \${DB_POSTGRESDB_USER} -d \${DB_POSTGRESDB_DATABASE}"]

interval: 10s

timeout: 5s

retries: 5

volumes:

traefik_data:

n8n_data:

db_data:

ANEXO D. INTEGRACIÓN CON SERVICIOS DE IA

Tabla 1

Integración con servicios de IA

Servicio	Tipo de integración	Funcionalidad verificada
Google Gemini	API REST con OAuth2 y clave API	Generación de texto, análisis de documentos, transcripción de audios, respuestas a preguntas.
OpenRouter	API REST con clave de API	Acceso a múltiples modelos de lenguaje de diferentes proveedores.

Servicio	Tipo de integración	Funcionalidad verificada
Google Sheets	OAuth2 y clave API	Lectura y escritura de datos tabulares, actualización de celdas.
Webhooks	HTTP/HTTPS	Disparadores automáticos de flujos desde aplicaciones externas
Gmail	OAuth2 y clave API	Envío automatizado de correos electrónicos con contenido dinámico

ANEXO E. COMPARATIVA DE ALTERNATIVAS

Tabla 2

Comparativa de alternativas

Criterio	n8n en VPS	n8n Cloud	Zapier/Make
Costo mensual	~\$20 USD	\$50+ USD	\$20-100+ USD
Control de datos	Total	Parcial	Limitado
Personalización	Alta	Media	Baja
Integración con IA local	Posible	No disponible	No disponible
Ejecuciones al mes	Ilimitadas	Limitadas por plan	Limitadas por plan
Soporte técnico	Comunidad	Incluido	Incluido