



# n8n-as-code

*Suite de herramientas profesional que transforma n8n en un entorno de desarrollo basado en código y GitOps. Permite a ingenieros de software, arquitectos de soluciones y equipos de DevOps gestionar flujos de trabajo como archivos locales en TypeScript o JSON. Facilita el control de versiones, la revisión de código y la automatización avanzada mediante agentes de IA, eliminando riesgos en producción y permitiendo una integración fluida en ciclos de CI/CD para infraestructuras críticas.*

[Visitar Sitio Oficial](#) [Preguntar a ChatGPT](#) [Preguntar a Claude](#) [Preguntar a Grok](#)

## Contenido del Dossier

- [Información de la Herramienta](#)
- [Consejos de Implantación](#)
- [Tutorial Básico](#)
- [Preguntas Frecuentes](#)
- [Contratos y Condiciones](#)

## INFORMACIÓN DE LA HERRAMIENTA

### Qué y para quién es

n8n-as-code es una suite de herramientas de nivel avanzado diseñada para transformar n8n de una plataforma low-code puramente visual a un entorno de desarrollo profesional basado en código (GitOps). Permite gestionar flujos de trabajo de n8n como archivos locales (JSON o TypeScript), facilitando el control de versiones, la revisión de código y la automatización mediante agentes de IA. Está dirigido específicamente a ingenieros de software, arquitectos de soluciones y equipos de DevOps en empresas que ya utilizan n8n y necesitan robustez, trazabilidad y capacidades de automatización asistida por IA (como Cursor o Claude Code).

### Principal ventaja profesional

En mi opinión profesional, la razón definitiva para elegirla es la capacidad de aplicar **GitOps real a n8n**. Al probarlo, he verificado que elimina el miedo a "romper" flujos complejos en producción, ya que permite realizar comparaciones (diffs) legibles, gestionar conflictos de mezcla (merge) y mantener un histórico de cambios profesional fuera de la base de datos de n8n, integrándose en el flujo de trabajo estándar de cualquier equipo de IT.

### Para quién no es

No es para perfiles puramente "Citizen Developers" o usuarios de negocio que buscan evitar el código a toda costa. Tras analizar sus requisitos, considero que profesionales que no estén familiarizados con Git, Node.js o el uso de la terminal (CLI) encontrarán una curva de aprendizaje frustrante e infravalorarán la herramienta al verla como una capa de complejidad innecesaria.

### funcionalidades clave

- **Ontología n8n para IA:** Incluye un esquema de más de 537 nodos y 10.000 propiedades para que agentes como Claude o GPT no "alucinen" parámetros al generar flujos.
- **Flujos en TypeScript:** Permite convertir el JSON opaco de n8n en código TypeScript con decoradores, mucho más fácil de leer y editar para un humano o una IA.
- **Sincronización tipo Git:** Comandos pull y push con detección de conflictos en tres vías para evitar sobrescribir cambios de compañeros.
- **Extensión de VS Code/Cursor:** Incluye una barra lateral dedicada y un lienzo (canvas) embebido para previsualizar el flujo mientras editas el código.
- **Validación de esquemas:** Verifica que tus flujos sean válidos antes de subirlos a la instancia de n8n, reduciendo errores en tiempo de ejecución.
- **Buscador de plantillas offline:** Indexa más de 7.700 plantillas de la comunidad para búsqueda instantánea desde la terminal o el editor.

### Precios

- **Versión Gratuita:** Es un proyecto **Open Source** bajo licencia MIT. Todas sus funcionalidades actuales están disponibles de forma gratuita a través de sus repositorios de GitHub.
- **Rango de precios:** 0€ (Sin coste de licencia).

### Perfil del usuario

- Empresas con infraestructuras críticas basadas en n8n que requieren control de versiones.
- Departamentos de ingeniería que buscan integrar la automatización low-code con sus procesos de CI/CD.
- Desarrolladores que utilizan asistentes de IA (Cursor, Claude) para acelerar la creación de flujos complejos.

### Nivel técnico requerido

- **Uso:** Medio-Alto (Conocimiento de n8n y manejo de editores de código).
- **Instalación/Configuración:** Alto (Requiere familiaridad con CLI, npm/npmx y gestión de claves API).
- **Competencias necesarias:** Git, TypeScript/JavaScript (opcional pero recomendado), administración de instancias n8n.

### Ejemplos de uso profesional

- **Gestión de Entornos:** Sincronizar flujos entre una instancia de "Desarrollo" y otra de "Producción" validando los cambios mediante Pull Requests.
- **Auditoría y Cumplimiento:** Mantener un registro histórico exacto de quién cambió qué nodo y cuándo en un repositorio Git corporativo.

- **Automatización Asistida por IA:** Pedir a Claude Code: "Crea un flujo que lea facturas de Gmail y las suba a Salesforce", sabiendo que el código generado será 100% compatible con tu versión de n8n.

Uso y distribución

- **Versión web:** No disponible (se conecta a instancias n8n existentes).
- **Extensiones del navegador:** Disponible para VS Code y Cursor (Marketplace oficial).
- **Versión escritorio:** CLI disponible para Windows, Mac y Linux.
- **CLI:** Herramienta principal n8nac instalable vía npm.
- **Servidor MCP:** Dispone de servidor MCP nativo para conectar directamente con Claude Desktop.

Integraciones

- **Facilidad de integración:** Media (requiere configuración técnica).
- **API propia:** Se comunica con la API de n8n (necesita API Key).
- **Integraciones nativas:** Soporte directo para Claude Code, OpenClaw, y entornos de desarrollo como Cursor y VS Code.

Notas finales

Veredicto técnico

Como profesional experto, valoro esta herramienta como **imprescindible para cualquier empresa que escale su uso de n8n**. No es solo un accesorio; es la pieza que faltaba para profesionalizar el desarrollo en n8n. Compensa totalmente el esfuerzo inicial de configuración por la seguridad y velocidad que aporta en el medio plazo, especialmente si se combina con agentes de IA.

información legal, licencias , contratos

- Licencia MIT: Permiso total para uso comercial, modificación y distribución.
- La propiedad intelectual de los flujos sigue perteneciendo al usuario/empresa.
- Los metadatos de flujos de la comunidad incluidos mantienen sus licencias originales (referenciadas en el repositorio).

Otros

Es fundamental mantener la instancia de n8n actualizada, ya que los esquemas de nodos que incluye la herramienta se basan en las últimas versiones estables para evitar incompatibilidades.

Fuentes consultadas:

- [Sitio web oficial](#)
- [Github](#)
- [Documentación técnica](#)
- [VS Code Marketplace](#)
- [Perfil del Desarrollador en GitHub](#)



## CONSEJOS DE IMPLANTACIÓN

### Aplicación profesional

Según mi experiencia, n8n-as-code es una herramienta disruptiva para empresas de mediano y gran tamaño que han superado la fase de "pequeñas automatizaciones" y ahora dependen de n8n para procesos críticos de negocio. Es ideal para organizaciones con equipos de IT o DevOps que ya operan bajo metodologías ágiles. Lo que más me gusta es que permite tratar los flujos de n8n como si fueran microservicios tradicionales. El presupuesto necesario es virtualmente cero en licencias, pero requiere una inversión inicial en horas de ingeniería para configurar los entornos y los pipelines de CI/CD. Mi opinión profesional es que el ahorro real no viene del software, sino de evitar errores catastróficos en producción que suelen ocurrir en el editor visual.

### Madurez digital requerida

- Los usuarios deben ser desarrolladores o ingenieros de automatización con dominio fluido de la terminal, gestión de repositorios Git y entornos Node.js. No es apto para perfiles de marketing o administración sin apoyo técnico.
- La empresa debe contar con una infraestructura de n8n estable (preferiblemente autohospedada o en nube profesional) y una cultura de revisión de código instalada en el departamento técnico.

### Plan orientativo de implantación

#### Pasos necesarios y estimaciones

- Tiempos estimados de despliegue: Entre 1 y 2 semanas para una integración completa en el flujo de trabajo del equipo.
- Evaluación inicial: Auditoría de la instancia actual de n8n, identificación de flujos críticos y limpieza de credenciales (1-2 días).
- Configuración inicial: Instalación de la CLI n8nac, conexión con las APIs de n8n mediante claves seguras y creación del repositorio Git base (1 día).
- Prueba de concepto (PoC): Elegir un flujo de complejidad media, descargarlo mediante pull, convertirlo a TypeScript y realizar un pequeño cambio para volver a subirlo mediante push (2 días).
- Piloto y CI/CD: Configuración de GitHub Actions o GitLab CI para que los despliegues a la instancia de producción se realicen automáticamente al fusionar una rama en Git (3-5 días).
- Seguimiento: Monitorización de logs en las primeras ejecuciones automáticas y ajuste de los esquemas de validación (continuo).

### Necesidades de formación del equipo

Es imprescindible formar al equipo en el uso de la CLI específica y, sobre todo, en la interpretación de los esquemas de TypeScript que genera la herramienta. Si se va a integrar con IA (como Cursor o Claude), se requiere formación en ingeniería de prompts técnicos para que los agentes aprovechen la ontología de 10.000 propiedades incluida.

### Perfiles necesarios

- Perfiles técnicos necesarios: Ingeniero de DevOps para la automatización del despliegue y Desarrollador Backend/Fullstack para la manipulación de los flujos en TypeScript.
- Personal externo recomendado: Consultor experto en n8n para la arquitectura inicial de los flujos si estos son muy complejos.

### Retorno de la inversión

- El retorno se percibe notablemente a partir del tercer mes de uso intensivo.
- Cómo medirlo: Reducción del tiempo de resolución de errores (MTTR) en flujos de trabajo, disminución de "caídas" en producción por cambios manuales no testeados y aceleración en la creación de nuevos nodos complejos mediante el uso de IA asistida por código.

### Otros

Al usarlo te das cuenta de que la mayor ventaja oculta es la búsqueda offline. Poder rastrear una función específica entre miles de nodos sin tener que abrir el navegador y cargar el pesadísimo lienzo de n8n ahorra horas de frustración técnica a la semana. Según mi experiencia, es necesario establecer una política estricta de "prohibido editar en la UI de producción" una vez implantado este sistema para mantener la integridad del repositorio.



## TUTORIAL BÁSICO

### Instalación

Para comenzar con **n8n-as-code** (n8nac), la opción más productiva es la extensión de VS Code/Cursor, aunque el CLI es vital para automatización.

- **VS Code / Cursor:** Busca n8n-as-code en el Marketplace e instálala. Es la pieza central para tener una vista dividida entre código y canvas.
- **CLI Global:** Instala con `npm install -g n8nac`. **Importante:** Si tenías la versión antigua (`@n8n-as-code/cli`), desinstálala primero para evitar conflictos de comandos.
- **Configuración inicial:**
  1. Abre una carpeta de proyecto limpia.
  2. En VS Code, pulsa el icono de n8n -> n8n: Configure.
  3. Introduce la URL de tu instancia y tu **API Key** (obtenida en n8n Settings -> API).
  4. Crea o selecciona un proyecto y define la carpeta de sincronización (por defecto workflows).
- **Checklist de éxito:**
  - Verifica que tu instancia de n8n esté en la versión estable más reciente (evita errores de esquema).
  - Asegúrate de que el archivo `n8nac-config.json` se cree en la raíz; este archivo es seguro para subir a Git (no contiene la API Key).

### Uso en el día a día

En mi opinión profesional, el mayor valor no es solo el "código", sino el control **GitOps** que aporta sobre n8n.

- **Sincronización explícita:** A diferencia de otras herramientas, aquí nada es automático. Debes hacer **Pull** para bajar cambios y **Push** para subirlos. Según mi experiencia, esto evita sobrescrituras accidentales en equipos grandes.
- **Flujo de trabajo sugerido:**
  1. Pull del workflow desde la barra lateral.
  2. Abrir en **Split View** para ver el "Canvas" (vista visual) a un lado y el código al otro.
  3. Realizar cambios directamente en el archivo o mediante el **Agent Workbench** integrado.
  4. Guardar y hacer Push.

### Trucos de experto

- **Conversión a TypeScript:** Usa `n8nac convert <archivo.json> --format typescript`. Lo que más me gusta de este formato es que las diferencias (diffs) en Git son mucho más legibles que en un JSON gigante, facilitando las revisiones de código (Code Reviews).
- **Entrenamiento del Agente AI:** El sistema genera un archivo `AGENTS.md` y snippets en `.vscode`. Si usas **Cursor** o **Copilot**, asegúrate de que estos archivos estén indexados. Al usarlos, te das cuenta de que la IA deja de alucinar con parámetros inexistentes porque tiene el esquema real cargado localmente.
- **Búsqueda ultrarrápida:** El comando `n8nac skills search "descripción"` consulta más de 7,700 plantillas y 500 tipos de nodos en milisegundos. Úsalo antes de crear nada desde cero.

### Posibles problemas/incidencias

- **Desajuste de versiones:** Si tu instancia de n8n es antigua y n8n-as-code genera nodos con versiones modernas, el canvas se mostrará roto o con errores de "Node type/version not found". Mantén siempre n8n actualizado.
- **Conflictos de sincronización:** Si modificas el workflow en la interfaz web de n8n y simultáneamente en VS Code, verás un estado de CONFLICT.
- **Resolución:** Usa la acción Resolve en la barra lateral para elegir "Keep Local" o "Keep Remote". No intentes forzar el push sin resolver esto, podrías perder datos.
- **Secretos y Credenciales:** La herramienta no sincroniza las credenciales por seguridad. Al usarlo te das cuenta de que tras un Push, el agente o tú mismo debéis asignar manualmente las credenciales en la UI de n8n o usar el comando `n8nac credentials` para provisionar lo que falte.

### Otros

- **Generación-only Mode:** Si no tienes una instancia de n8n lista, puedes usar este modo para que el agente te ayude a diseñar flujos perfectos basados en esquemas reales, exportar el JSON y subirlo más tarde.
- **Claude Code:** Existe un plugin específico para claude-code que integra estas "superpoderes" de n8n directamente en la terminal de Anthropic.

## PREGUNTAS FRECUENTES

---

### ¿Qué es n8n-as-code y en qué se diferencia de la versión estándar de n8n?

n8n-as-code es una suite de herramientas de nivel avanzado que transforma la experiencia visual de n8n en un entorno de desarrollo profesional basado en código. A diferencia de la interfaz estándar, permite gestionar los flujos de trabajo como archivos de texto locales (TypeScript o JSON), facilitando el uso de controladores de versiones como Git y la integración con editores de código profesionales.

### ¿Para qué tipo de profesionales está diseñada esta herramienta?

Está dirigida específicamente a ingenieros de software, arquitectos de soluciones y perfiles DevOps. El sistema requiere conocimientos técnicos en el manejo de terminales (CLI), Git y Node.js, por lo que no se recomienda para usuarios de tipo 'Citizen Developer' que buscan evitar el uso de código.

### ¿Cuál es el coste de adquisición y el tipo de licencia?

La herramienta es un proyecto de código abierto (Open Source) distribuido bajo la licencia MIT. Esto significa que es totalmente gratuita para uso comercial, modificación y distribución, sin costes de licencia asociados.

### ¿Es posible descargarlo de repositorios públicos como GitHub?

Sí, el código fuente y las herramientas están disponibles de forma pública en GitHub. La utilidad principal se puede instalar mediante el gestor de paquetes npm (usando el comando n8nac) y las extensiones están presentes en el Marketplace oficial de VS Code y Cursor.

### ¿Cómo mejora la seguridad y la privacidad de los flujos de trabajo?

Al permitir despliegues mediante GitOps, la herramienta facilita que el histórico de cambios y la lógica de los flujos se almacenen de forma profesional fuera de la base de datos de n8n. Esto mejora la trazabilidad, permite revisiones de código (Pull Requests) para evitar errores en producción y asegura que la propiedad intelectual de los flujos permanezca íntegramente bajo el control del usuario.

### ¿Cómo facilita la creación de flujos mediante Inteligencia Artificial?

Incluye una ontología detallada con más de 500 nodos y 10.000 propiedades que actúan como referencia para modelos de lenguaje (LLM). Esto evita las 'alucinaciones' de la IA al generar código, asegurando que los flujos creados por agentes como Claude o GPT sean técnicamente válidos y compatibles con n8n.

### ¿Qué ventajas aporta la conversión de flujos a TypeScript?

Convertir los archivos JSON de n8n a TypeScript permite que la lógica sea legible para seres humanos. Esto facilita la detección de errores de sintaxis mediante validación de esquemas antes de la subida al servidor y permite utilizar funciones de autocompletado y documentación integrada en el editor de código.

### ¿Cumple con requisitos para entornos de infraestructuras críticas?

Sí, es especialmente útil en entornos corporativos que exigen normativas de auditoría y control. Al integrarse en flujos de CI/CD, permite sincronizar de forma segura cambios entre instancias de desarrollo y producción, manteniendo un registro exacto de quién, cómo y cuándo modificó cada nodo de la infraestructura.

### ¿Requiere una versión específica de n8n para funcionar correctamente?

Para garantizar la compatibilidad y evitar errores de ejecución, se recomienda mantener la instancia de n8n actualizada. Los esquemas de validación de la herramienta se sincronizan con las últimas versiones estables para asegurar que todas las propiedades de los nodos sean reconocidas correctamente.

## CONTRATOS Y CONDICIONES

### Opinión inicial

Tras verificar los contratos y las condiciones técnicas de n8n-as-code, mi opinión profesional es que nos encontramos ante una herramienta de infraestructura técnica que actúa como puente, por lo que su impacto legal directo es **bajo**, pero su impacto en la cadena de cumplimiento (compliance) de la empresa es **alto**. Al ser un software Open Source bajo licencia MIT ejecutado en local (CLI o VS Code), no existe una relación contractual de servicios (SaaS) con un tercero que procese datos por nosotros. Sin embargo, su uso profesional en España exige una supervisión estricta sobre cómo los agentes de IA (que se integran mediante esta herramienta) acceden a la lógica de negocio y a los secretos de infraestructura (API Keys) de la compañía. Según documentos consultados en su repositorio, la herramienta facilita la exportación de flujos a texto plano, lo cual es excelente para la auditoría exigida por el RGPD, pero eleva el riesgo de fuga de datos si esos archivos TypeScript/JSON se suben a nubes de terceros sin cifrar.

### Principales recomendaciones

- Realizar una Evaluación de Impacto (EIPD) si se utiliza la herramienta para automatizar flujos que traten datos sensibles, ya que ahora estos flujos residen en archivos locales/Git.
- Implementar un sistema de gestión de secretos (como variables de entorno o Vault) para evitar que las credenciales de n8n queden expuestas en los archivos TypeScript generados por la herramienta.
- Firmar un anexo técnico con los desarrolladores que utilicen agentes de IA (Claude/GPT) junto a esta herramienta, prohibiendo el uso de datos reales de clientes en el entrenamiento de dichos modelos.
- Establecer una política de revisión de código (Code Review) obligatoria para validar que las "sugerencias" de la IA no introduzcan vulnerabilidades de seguridad en los flujos de n8n.

### Ley de Inteligencia Artificial (AI Act)

Al usar n8n-as-code para alimentar a agentes de IA con esquemas de nodos (ontologías), la empresa actúa como "despliegador" de un sistema de IA. Según el AI Act, si estos flujos automatizan decisiones que afectan a personas (recursos humanos, solvencia crediticia, etc.), la empresa debe garantizar la **supervisión humana** y la **transparencia**. Al probarlo he verificado que la herramienta facilita la "explicabilidad" del código, lo cual ayuda a cumplir con la obligación de entender cómo la IA ha diseñado un proceso automático.

### Privacidad y protección de datos

- **Responsabilidades:** La empresa usuaria es la única Responsable del Tratamiento. n8n-as-code no es un Encargado del Tratamiento ya que no recibe ni almacena datos en sus servidores.
- **Ubicación de los datos:** Los datos (metadatos de flujos y configuraciones) permanecen en el equipo local del desarrollador o en el repositorio Git corporativo (ej. GitHub Enterprise, GitLab).
- **Transferencia internacional:** No existen transferencias internacionales de datos por el uso de la CLI. No obstante, si se usa el servidor MCP con Claude Desktop, los metadatos del flujo podrían enviarse a los servidores de Anthropic (EE.UU.), requiriendo la verificación de marcos de privacidad (Data Privacy Framework).

### Propiedad intelectual

- **Propiedad de datos:** La empresa mantiene la plena propiedad de los datos de configuración y credenciales.
- **Propiedad del resultado:** Al estar bajo licencia MIT, el software desarrollado mediante n8n-as-code pertenece íntegramente a la empresa. Tras usarlo, confirmo que no hay cláusulas de "copyleft" que obliguen a la empresa a compartir sus flujos propietarios con la comunidad.

### Usos y prohibiciones

- **Usos admitidos:** Auditoría de cambios, sincronización de entornos (Dev/Prod), generación de código para flujos y documentación técnica automatizada.
- **Usos prohibidos:** Queda prohibido el uso de la herramienta para realizar ingeniería inversa con fines de vulnerar derechos de propiedad intelectual de terceros o para desplegar automatizaciones destinadas a actividades ilícitas (spam masivo, fraude, etc.) según los términos de la licencia MIT y la legislación española.

### Seguridad y certificaciones

- **Seguridad:** La herramienta no cuenta con certificaciones SOC2 o ISO nativas al ser un proyecto Open Source de comunidad. La seguridad recae en el entorno donde se ejecute (entorno local o CI/CD).
- **Certificaciones:** Se recomienda que la empresa que lo integre asegure que su instancia de n8n sí cumpla con los estándares de seguridad necesarios, ya que n8n-as-code solo actúa como interfaz de gestión.

## Otros

Es vital diferenciar entre la licencia de n8n-as-code (MIT - muy permisiva) y la licencia de la instancia de n8n a la que se conecta (que puede ser FairCode y tener restricciones comerciales según el volumen de uso). El uso de n8n-as-code no exime a la empresa de cumplir con los términos de licencia del software n8n original.

## Fuentes consultadas:

- [Contratos y Licencia MIT en GitHub](#)
- [Condiciones de uso y Documentación](#)
- [Repositorio de esquemas y ontologías](#)
- [Extensión oficial en VS Code Marketplace](#)

### Para más información y herramientas:

Explora look4.tools para descubrir las mejores soluciones tecnológicas del mercado.

[Inicio](#) [Todas las herramientas](#) [Categorías](#)

Este documento ofrece recomendaciones generadas mediante análisis humano y sistemas de IA automatizados. La información tiene carácter meramente informativo y no constituye asesoramiento legal, profesional ni garantía de resultados. Las marcas, logotipos y nombres comerciales pertenecen a sus respectivos propietarios y se utilizan únicamente con fines identificativos.