

# IACADEMY

DE FUNDAMENTOS A ARQUITECTURA DE IA

MÓDULO 21

## Automatización con n8n

Avanzado

iacademy.com — 2026

## MÓDULO 21

# Automatización con n8n

---

**Nivel:** Avanzado

**Autor:** Ricardo Gutierrez

**Publicación:** Mayo 2026

**Plataforma:** [iacademy.com](https://iacademy.com)

Este material es parte del curso completo de IAcademy.

Uso personal e intransferible. Queda prohibida su redistribución o reproducción sin autorización.

## Por que n8n y no otra herramienta

El mercado de automatizacion esta lleno de opciones: Zapier, Make, Power Automate, Tray.io. Todas funcionan. Pero n8n tiene una combinacion de ventajas que la convierte en la mejor opcion para profesionales y pymes que quieren control real sobre sus automatizaciones.

### Open source y transparente

El codigo de n8n esta en GitHub con licencia fair-code. Puedes revisarlo, entender como funciona, y contribuir. Esto no es solo filosofia: significa que si n8n desaparece manana, tu puedes seguir usando el software. Con Zapier o Make, si cierran o cambian los precios, pierdes todo.

### Self-hosted: tus datos en tu servidor

Puedes instalar n8n en tu propio servidor. Tus workflows, tus credenciales y tus datos no salen de tu infraestructura. Para cualquier empresa europea con requisitos de RGPD, esto es fundamental. Para sectores regulados (salud, finanzas, gobierno), es obligatorio.

### Sin limites de ejecucion

Zapier cobra por "tarea" (cada paso de un workflow cuenta). Un workflow de 5 pasos que se ejecuta 1000 veces al mes son 5000 tareas. En el plan Pro de Zapier (49 USD/mes) tienes 2000 tareas. En n8n self-hosted, el limite es la capacidad de tu servidor. Un VPS de 10 EUR/mes maneja miles de ejecuciones diarias sin problema.

### Flexibilidad total con codigo

n8n tiene un nodo de "Code" donde puedes escribir JavaScript o Python directamente. Si una integracion no existe como nodo nativo, la creas con un HTTP Request o un bloque de codigo en 10 minutos. En Zapier, si no hay conector, estas atascado.

## Comparativa directa

Característica	n8n (self-hosted)	Zapier	Make
Precio base	Gratis (+ coste VPS)	19.99 USD/mes	9 EUR/mes
Ejecuciones	Ilimitadas	750/mes (basico)	1000/mes (basico)
Self-hosted	Si	No	No
Nodo de codigo	JS + Python	Limitado	JS basico
Integraciones	400+	6000+	1500+
Open source	Si (fair-code)	No	No
Soberania datos	Total	Ninguna	Ninguna

Zapier gana en cantidad de integraciones nativas. Pero n8n gana en flexibilidad, coste a escala y control. Para un profesional que construye automatizaciones serias, n8n es la eleccion correcta.

## Instalacion con Docker

La forma recomendada de instalar n8n es con Docker Compose. Necesitas un servidor (Hetzner, DigitalOcean, cualquier VPS con Docker) y 5 minutos.

### Docker Compose basico

```
version: '3.8'

services:
  n8n:
    image: n8nio/n8n:latest
    restart: unless-stopped
    ports:
      - "5678:5678"
    environment:
      - N8N_HOST=n8n.tudominio.com
      - N8N_PORT=5678
      - N8N_PROTOCOL=https
```

```

- WEBHOOK_URL=https://n8n.tudominio.com/
- GENERIC_TIMEZONE=Europe/Madrid
- TZ=Europe/Madrid
volumes:
- n8n_data:/home/node/.n8n

volumes:
n8n_data:

```

Ejecuta `docker compose up -d` y n8n estara corriendo en el puerto 5678. La primera vez que accedas, te pedira crear un usuario administrador.

## Configuracion con PostgreSQL (produccion)

Para produccion, n8n deberia usar PostgreSQL en vez de SQLite (el default). Esto mejora el rendimiento con muchos workflows y permite backups mas robustos:

```

services:
  n8n:
    image: n8nio/n8n:latest
    restart: unless-stopped
    ports:
      - "5678:5678"
    environment:
      - DB_TYPE=postgresdb
      - DB_POSTGRESDB_HOST=postgres
      - DB_POSTGRESDB_PORT=5432
      - DB_POSTGRESDB_DATABASE=n8n
      - DB_POSTGRESDB_USER=n8n
      - DB_POSTGRESDB_PASSWORD=${N8N_DB_PASSWORD}
      - N8N_HOST=n8n.tudominio.com
      - WEBHOOK_URL=https://n8n.tudominio.com/
      - GENERIC_TIMEZONE=Europe/Madrid
    volumes:
      - n8n_data:/home/node/.n8n
    depends_on:
      - postgres

  postgres:
    image: postgres:16-alpine
    restart: unless-stopped
    environment:
      - POSTGRES_USER=n8n

```

```

- POSTGRES_PASSWORD=${N8N_DB_PASSWORD}
- POSTGRES_DB=n8n
volumes:
- postgres_data:/var/lib/postgresql/data

volumes:
  n8n_data:
  postgres_data:

```

## Opcion cloud

Si no quieres gestionar infraestructura, n8n.io ofrece un plan cloud desde 20 EUR/mes. Incluye hosting, actualizaciones y soporte. Para empezar esta bien, pero a largo plazo el self-hosted es mas economico y te da mas control.

### Decision rapida

- **Tienes VPS o Hetzner:** self-hosted siempre.
- **No quieres gestionar servidor:** n8n cloud.
- **Datos sensibles o regulacion:** self-hosted obligatorio.

## Conceptos clave: workflows, nodos, triggers

### Workflows

Un workflow es una secuencia de pasos que se ejecutan en orden. Cada workflow tiene un nombre, un estado (activo/inactivo), y un historial de ejecuciones. Puedes tener decenas de workflows activos simultaneamente, cada uno haciendo una tarea diferente.

### Nodos

Cada paso del workflow es un nodo. Existen tres tipos principales:

- **Nodos de accion:** hacen algo (enviar email, insertar en base de datos, llamar a una API).
- **Nodos de logica:** toman decisiones (If/Else, Switch, Merge, Split).

- **Nodos de transformacion:** manipulan datos (Set, Code, Function, Item Lists).

## Triggers

Un trigger es el nodo que inicia el workflow. Sin trigger, el workflow solo se ejecuta manualmente. Tipos de triggers:

- **Schedule Trigger:** ejecuta en un horario definido (cada hora, cada lunes a las 8:00, etc.).
- **Webhook:** se ejecuta cuando alguien llama a una URL especifica.
- **Event Triggers:** se ejecuta cuando pasa algo en un servicio (email recibido, push en GitHub, pago en Stripe).
- **Error Trigger:** se ejecuta cuando otro workflow falla.

## Credenciales

Las credenciales almacenan las claves de acceso a servicios externos (API keys, tokens OAuth, passwords). Se guardan cifradas en la base de datos de n8n y se referencian por nombre en los nodos. Nunca hardcodees una API key directamente en un nodo.

## Expresiones

Las expresiones permiten referenciar datos de nodos anteriores. Si el nodo "Webhook" recibe un JSON con un campo "email", en el nodo siguiente puedes acceder a ese valor con `{{ $json.email }}`. Es el sistema que conecta la salida de un nodo con la entrada del siguiente.

# 10 workflows esenciales

Estos 10 workflows cubren el 90% de las necesidades de automatizacion de un profesional o pyme. Cada uno incluye los nodos necesarios, la configuracion clave y las consideraciones practicas.

## 1. RSS feed a newsletter con IA

**Nodos:** Schedule Trigger → RSS Read → HTTP Request (Claude API) → Brevo (Send Email)

**Que hace:** cada lunes a las 8:00 AM, lee los últimos 5 artículos de un feed RSS de tu sector. Envía cada artículo a la API de Claude con el prompt: "Resume este artículo en 3 bullets de máximo 50 palabras cada uno, tono profesional, en español." Con los resúmenes, compone una newsletter y la envía vía Brevo a tu lista de suscriptores.

**Configuración clave:** en el nodo RSS, limita a 5 items para no exceder tokens. En Claude, usa claude-haiku-4-20250414 para resumen rápido y barato. En Brevo, configura un template de newsletter con variables para insertar los resúmenes.

## 2. Push en GitHub a Slack

**Nodos:** GitHub Trigger (push) → Set (formatear) → Slack (Send Message)

**Que hace:** cada vez que alguien hace push a un repositorio, envía un mensaje a un canal de Slack con el autor, mensaje del commit, rama y archivos modificados.

**Configuración clave:** configura el GitHub webhook apuntando a tu n8n. En el nodo Set, formatea el mensaje con Markdown para Slack. Filtra por rama si solo quieres notificaciones de main/master.

## 3. Informe diario de PostgreSQL a Telegram

**Nodos:** Schedule Trigger (9:00 AM) → PostgreSQL (Execute Query) → Set (formatear) → Telegram (Send Message)

**Que hace:** cada mañana ejecuta una query SQL (nuevos registros, métricas, errores del día anterior) y envía un resumen formateado a un chat de Telegram.

**Configuración clave:** la query SQL debe ser específica y limitada (no hagas SELECT \* FROM tabla). Formatea el mensaje con emojis y secciones para que sea legible en el móvil.

## 4. Webhook de validación a Supabase

**Nodos:** Webhook → If (validar campos) → Supabase (Insert) / Respond to Webhook (error 400)

**Que hace:** recibe datos de un formulario web. Valida que los campos obligatorios existan, que el email tenga formato correcto y que no haya datos maliciosos. Si todo está bien, inserta en Supabase. Si no, devuelve un error.

**Configuración clave:** usa expresiones en el nodo If para validar:

`{{ $json.email.includes('@') }}`. Configura el webhook para responder inmediatamente (modo "Respond to Webhook" como último nodo).

## 5. Clasificar emails con Claude

**Nodos:** IMAP Trigger → HTTP Request (Claude API) → Switch → Brevo (Add to List) / Slack (Notify)

**Que hace:** cuando llega un email, extrae asunto y cuerpo. Claude lo clasifica en categorías (urgente, soporte, comercial, spam). Según la categoría, lo enruta: urgentes van a Slack, soporte va a una lista de Brevo, comerciales a otra lista, spam se descarta.

**Configuración clave:** el prompt a Claude debe ser muy específico: "Clasifica este email en exactamente una de estas categorías: urgente, soporte, comercial, spam. Responde SOLO con la categoría, sin explicación." Usa el nodo Switch con condiciones exactas sobre la respuesta.

## 6. Scraping de licitaciones públicas (PLACSP)

**Nodos:** Schedule Trigger (8:00 AM) → HTTP Request (PLACSP) → Code (parsear HTML) → HTTP Request (Claude API) → If (relevante) → Telegram

**Que hace:** cada día consulta la Plataforma de Contratación del Sector Público, extrae las licitaciones nuevas, Claude las clasifica por relevancia para tu sector, y las relevantes se envían por Telegram con enlace directo.

**Configuración clave:** el nodo Code parsea el HTML de la PLACSP para extraer título, organismo, presupuesto y enlace. El prompt de Claude incluye tu sector y criterios de relevancia. Filtra por presupuesto mínimo para evitar ruido.

## 7. Stripe webhook a Supabase y email de bienvenida

**Nodos:** Webhook (Stripe) → Supabase (Update) → Brevo (Send Email)

**Que hace:** cuando un cliente paga en Stripe, el webhook llega a n8n. Actualiza el campo "plan" y "payment\_date" en Supabase. Envía un email de bienvenida personalizado con el nombre del cliente, su plan y los próximos pasos.

**Configuración clave:** configura el webhook de Stripe para eventos "checkout.session.completed". Verifica la firma del webhook para seguridad. En Brevo, usa un template transaccional con variables.

## 8. Formulario a contacto Brevo con enlace PDF

**Nodos:** Webhook → Brevo (Create Contact) → Code (generar enlace firmado) → Brevo (Send Email)

**Que hace:** un formulario de lead magnet envía datos a n8n. Crea el contacto en Brevo y lo asigna a una lista. Genera un enlace de descarga firmado (con expiración de 48 horas) al PDF. Envía el email con el enlace.

**Configuración clave:** el enlace firmado evita que el PDF se comparta sin control. Usa un nodo Code para generar un token temporal con crypto. El contacto se crea antes de enviar el email para que entre en la secuencia de email marketing.

## 9. Métricas diarias de PostHog a Telegram

**Nodos:** Schedule Trigger (9:30 AM) → HTTP Request (PostHog API) → Code (calcular métricas) → Telegram

**Que hace:** consulta la API de PostHog para obtener visitantes únicos, eventos clave y tasas de conversión del día anterior. Calcula variaciones respecto al día anterior. Envía un resumen a Telegram.

**Configuración clave:** la API de PostHog requiere un project API key y la query en formato HogQL. El nodo Code calcula los deltas y formatea el mensaje con flechas arriba/abajo según la tendencia.

## 10. Alerta de errores en cualquier workflow

**Nodos:** Error Trigger → Set (formatear) → Telegram (Send Message)

**Que hace:** cuando cualquier workflow falla, este meta-workflow captura el nombre del workflow fallido, el mensaje de error, el nodo donde falló y el timestamp. Envía una alerta inmediata a Telegram.

**Configuración clave:** el Error Trigger es un tipo especial de trigger que se activa ante fallos de otros workflows. Incluye la URL directa al workflow fallido para que puedas ir a depurar con un click.

## Gestion de credenciales

Las credenciales son el punto mas sensible de n8n. Una mala gestion puede exponer API keys, tokens y passwords.

### Reglas de credenciales

1. **Nunca hardcodees.** Usa siempre el sistema de credenciales nativo de n8n. Se cifran en la base de datos.
2. **Variables de entorno para produccion.** Define tus credenciales como variables de entorno en docker-compose y referencia con `${VARIABLE}` en n8n.
3. **Credenciales separadas por entorno.** Credenciales de desarrollo y produccion deben ser diferentes. Nunca uses la API key de produccion para pruebas.
4. **Rotacion periodica.** Cambia las API keys cada 90 dias minimo. n8n permite actualizar credenciales sin tocar los workflows.
5. **Permisos minimos.** Si un workflow solo necesita leer de una base de datos, la credencial debe tener solo permisos de lectura.

## Error handling y buenas practicas

Un workflow sin error handling es una bomba de relojeria. Tarde o temprano un API va a devolver un error, un servicio va a estar caido, o un dato va a tener un formato inesperado.

### Patron de error handling

1. **Retry:** configura cada nodo critico con 3 reintentos y backoff exponencial (espera 1s, 2s, 4s).
2. **Fallback:** si despues de los reintentos sigue fallando, el flujo va a un nodo alternativo (por ejemplo, guardar en una cola para procesamiento posterior).
3. **Alerta:** siempre notifica cuando algo falla. El workflow 10 (Error Trigger a Telegram) es obligatorio.
4. **Log:** guarda los errores en una base de datos o archivo para analisis posterior.

## Buenas practicas generales

- Nombra tus nodos con nombres descriptivos, no "HTTP Request 1". Usa "Claude: clasificar email" o "Brevo: enviar newsletter".
- Anade notas (sticky notes) a los workflows complejos explicando la logica.
- Limita el volumen de datos por ejecucion. Si procesas 10.000 items, divide en lotes.
- Testea cada nodo individualmente antes de activar el workflow completo.
- Usa el modo de ejecucion manual para depurar antes de activar el trigger automatico.

## Nodo Claude API en detalle

Claude no tiene un nodo nativo en n8n (a diferencia de OpenAI). Se conecta mediante un nodo HTTP Request configurado manualmente. Esto da mas control sobre los parametros.

### Configuracion del nodo

```
Metodo: POST
URL: https://api.anthropic.com/v1/messages

Headers:
  x-api-key: {{ $credentials.claudeApiKey }}
  anthropic-version: 2023-06-01
  content-type: application/json

Body (JSON):
{
  "model": "claude-sonnet-4-20250514",
  "max_tokens": 1024,
  "messages": [
    {
      "role": "user",
      "content": "{{ $json.prompt }}"
    }
  ],
  "system": "Eres un asistente profesional. Responde en español, de forma concisa."
}
```

## Elegir el modelo correcto

- **claude-haiku-4-20250414:** rapido y barato. Para clasificacion, extraccion de datos, respuestas cortas. Coste: ~0.25 USD por millon de tokens de entrada.
- **claude-sonnet-4-20250514:** balance calidad/coste. Para resumen, redaccion, analisis. El modelo por defecto para la mayoria de workflows.
- **claude-opus-4-20250514:** maxima calidad. Para tareas complejas que requieren razonamiento profundo. Coste elevado, usar solo cuando es necesario.

## Parsear la respuesta

La respuesta de Claude viene en JSON. El texto esta en `content[0].text`. Usa un nodo Set despues del HTTP Request para extraer solo el texto:

```
{{ $json.content[0].text }}
```

## Gestion de tokens

Cada llamada a Claude consume tokens (y dinero). Para optimizar:

- Limita el contexto: no envíes textos completos si solo necesitas un fragmento.
- Usa `max_tokens` ajustado: si esperas una respuesta de 50 palabras, pon `max_tokens` en 200, no en 4096.
- Cachea resultados: si el mismo input produce el mismo output, guarda la respuesta y reutilízala.

## Nodo Brevo: contactos, listas y emails

Brevo (antes Sendinblue) tiene nodo nativo en n8n. Las operaciones mas utiles:

### Crear contacto

Crea un nuevo contacto con email, nombre, y atributos personalizados. Asignalo a una o mas listas. Util para lead magnets, formularios de registro y webhooks de pago.

## Enviar email transaccional

Envia un email individual usando un template de Brevo. Los templates se crean en el dashboard de Brevo con variables ({{ params.nombre }}, {{ params.enlace }}). n8n pasa los valores de las variables al enviar.

## Enviar campana

Dispara una campana de email a una lista completa. Util para newsletters automatizadas. Crea la campana en Brevo con el contenido y usa n8n para dispararla en el momento correcto.

## Automation triggers

Brevo tiene sus propias automatizaciones. Puedes usar n8n para anadir un contacto a una lista que dispara una secuencia automatica de emails en Brevo. Asi combinas la logica de n8n con las secuencias de Brevo.

## Best practices de self-hosting

### Backups

Si usas PostgreSQL, configura pg\_dump diario:

```
# Cron: todos los dias a las 3:00 AM
0 3 * * * docker exec n8n-postgres pg_dump -U n8n n8n | gzip > /backups/n8n_$(date +%Y%m%d).sql.gz
```

Guarda los backups en un almacenamiento externo (S3, otro servidor). Retiene 30 dias y elimina los antiguos.

### Actualizaciones

n8n publica actualizaciones frecuentes. El proceso:

1. Lee el changelog antes de actualizar.
2. Haz backup de la base de datos.
3. `docker compose pull` para descargar la nueva imagen.

4. `docker compose up -d` para reiniciar con la nueva version.
5. Verifica que los workflows activos siguen funcionando.

## Monitoring

Configura un health check externo (UptimeRobot, Healthchecks.io) que haga ping a tu instancia de n8n cada 5 minutos. Si n8n se cae, recibes una alerta inmediata. Complementa con el workflow 10 (Error Trigger) para errores internos.

## Seguridad

- Pon n8n detras de un reverse proxy (Caddy, Traefik) con HTTPS obligatorio.
- Restringe el acceso al puerto 5678 por IP si es posible.
- Usa autenticacion fuerte: password largo y unico.
- Desactiva el registro publico: `N8N_USER_MANAGEMENT_DISABLED=true` (despues de crear tu usuario).

## Ejercicio practico: RSS a newsletter con Claude

### Ejercicio M21: Construye tu primer workflow automatizado

1. **Instala n8n.** Si tienes Docker, usa el docker-compose basico. Si no, usa n8n.cloud (plan gratuito).
2. **Crea un nuevo workflow** llamado "Newsletter semanal con IA".
3. **Anade Schedule Trigger:** configuralo para cada lunes a las 8:00 AM (Europe/Madrid).
4. **Anade RSS Read:** URL de un feed de tu sector. Limita a 5 items.
5. **Anade HTTP Request (Claude API):** configura POST a la API de Claude. Modelo: claude-haiku-4-20250414. Prompt: "Resume este articulo en 3 bullets de maximo 50 palabras cada uno. Tono profesional. En espanol."
6. **Anade nodo Brevo:** Send Email con tu template de newsletter. Mapea los resúmenes a las variables del template.
7. **Anade Error Trigger** → **Telegram** en un workflow separado para recibir alertas si falla.
8. **Testea manualmente** cada nodo. Verifica los datos entre nodos.

9. **Activa el workflow** y espera al lunes para ver el resultado real.

**Bonus:** Añade un segundo workflow del listado (elige el que mas te sirva) y tenlo funcionando esta semana.

## Conclusiones clave

### Key takeaways del M21

1. n8n es open source, self-hosted y sin limites de ejecucion. La mejor opcion para profesionales que quieren control total.
2. Los 5 conceptos clave: workflows, nodos, triggers, credenciales y expresiones. Dominalos y puedes automatizar cualquier cosa.
3. Los 10 workflows esenciales cubren el 90% de las necesidades: newsletter con IA, notificaciones, informes, validacion, clasificacion, scraping, pagos, lead magnets, metricas y alertas de error.
4. Nunca hardcodees credenciales. Usa el sistema nativo de n8n y variables de entorno para produccion.
5. Error handling no es opcional: retry, fallback, alerta. Sin esto, tus workflows fallaran en silencio.
6. Claude API se conecta via HTTP Request. Elige el modelo segun la tarea: Haiku para clasificacion, Sonnet para redaccion, Opus para razonamiento complejo.

## Siguiente: M22 - Gamma + herramientas visuales con IA

Has automatizado procesos. Ahora toca que lo que produces se vea profesional: Gamma, Canva y Figma con IA.

[Ir al Modulo 22](#)

# IACADEMY

[iacademy.com](https://iacademy.com)

---

De fundamentos a arquitectura de IA.  
12 módulos prácticos. 24 recursos descargables.  
Quizzes con certificado. Vídeos profesionales.

Empieza gratis en [iacademy.com/free](https://iacademy.com/free)

© 2026 IAcademy — Todos los derechos reservados