



En este módulo

1. Qué es un workflow automatizado
2. Identificar tareas automatizables
3. Herramientas de automatización
4. n8n: automatización visual
5. Claude Code hooks
6. Zapier y Make: sin código
7. 5 workflows que deberías automatizar
8. Errores al automatizar
9. Escalar automatizaciones

Qué es un workflow automatizado

Un workflow automatizado es una secuencia de pasos que se ejecutan sin intervención manual. En lugar de hacer una tarea repetitiva paso a paso, defines la secuencia una vez y la máquina la ejecuta cada vez que se necesita.

Ejemplo: Workflow de contenido para redes sociales

MANUAL (45 min/día):

1. Buscar noticias del sector
2. Seleccionar las 3 más relevantes
3. Escribir un post para LinkedIn
4. Crear una imagen
5. Publicar en el horario óptimo

AUTOMATIZADO (0 min/día):

Trigger: Cada día a las 7:00

1. RSS → Recopilar noticias de 10 fuentes
2. IA → Seleccionar y resumir las 3 mejores
3. IA → Generar post adaptado a LinkedIn
4. API → Generar imagen con plantilla
5. API → Programar publicación a las 9:00

Todo workflow tiene la misma anatomía: un disparador que decide cuándo empieza, una serie de pasos que transforman los datos y un destino donde queda el resultado. En el ejemplo de redes

sociales, el disparador es la hora, los pasos son leer noticias, seleccionarlás y redactar, y el destino es la publicación programada. Cuando analices cualquier tarea, intenta describirla con esas tres piezas; si no sabes decir qué la dispara o dónde termina, todavía no está lista para automatizarse.

Fíjate también en qué hace la IA dentro del flujo: seleccionar, resumir y redactar, es decir, las partes que antes exigían leer y escribir. El resto son conexiones entre servicios. El error típico es creer que automatizado significa sin revisión. Aunque el tiempo diario baje a casi cero, alguien debe comprobar periódicamente que lo publicado sigue teniendo sentido.

Identificar tareas automatizables

No todo se puede ni se debe automatizar. Usa esta matriz:

AUTOMATIZAR SI:

- ✓ Se repite más de 3 veces por semana
- ✓ Sigue pasos predecibles
- ✓ No requiere juicio humano crítico
- ✓ El error es fácilmente detectable
- ✓ Te lleva más de 10 minutos cada vez

NO AUTOMATIZAR SI:

- ✗ Requiere decisión estratégica
- ✗ Involucra datos sensibles sin supervisión
- ✗ Un error tiene consecuencias graves
- ✗ Se hace menos de 1 vez al mes
- ✗ El proceso cambia constantemente

Regla 80/20

Automatiza el 80% de la tarea y deja el 20% para revisión humana. Es más seguro, más rápido de implementar y te da control sobre el resultado final.

La matriz se entiende mejor con números. Supongamos, con datos inventados, que cada mañana dedicas doce minutos a copiar los pedidos nuevos de la tienda a una hoja de cálculo, cinco días por semana. Son unas cuatro horas al mes de una tarea predecible, repetitiva y cuyo error se detecta fácilmente comparando totales. Si montar la automatización te lleva tres horas, la recuperas en menos de un mes.

La regla del ochenta veinte aplicada a ese ejemplo significa que la máquina copia y valida los pedidos, y tú solo revisas los que llegan con algún campo raro. El error típico es elegir la tarea más vistosa en lugar de la más frecuente: automatizar algo espectacular que haces dos veces al año ahorra mucho menos que automatizar algo aburrido que haces cada día.

Herramientas de automatización

- **n8n:** Open-source, self-hosted, con nodo de IA integrado. La opción más flexible.
- **Zapier:** El más fácil de usar. Sin código. Miles de integraciones. Plan gratuito limitado.
- **Make (Integromat):** Más potente que Zapier, interfaz visual, mejor precio por automatización.
- **Claude Code hooks:** Scripts que se ejecutan automáticamente antes o después de acciones.
- **GitHub Actions:** Automatización de flujos de desarrollo (CI/CD, tests, deploys).

Para elegir entre estas opciones, hazte tres preguntas. La primera, dónde deben vivir los datos: si no pueden salir de tu infraestructura, necesitas algo que puedas alojar tú, como n8n. La segunda, quién va a mantener el flujo: si es una persona sin perfil técnico, una herramienta sin código reduce la dependencia de desarrolladores. La tercera, dónde ocurre la tarea: si sucede mientras programas con Claude Code, los hooks son el sitio natural, y si forma parte del ciclo de un repositorio, como pasar tests antes de fusionar cambios, GitHub Actions encaja mejor.

El error típico es elegir por moda y descubrir después que la herramienta no tiene la integración que necesitabas. Antes de decidir, comprueba que existen conectores para los servicios concretos que vas a usar.

n8n: automatización visual

n8n es una herramienta de automatización visual que conecta servicios con nodos de IA.

Instalar n8n:

```
npm install -g n8n
n8n start
```

Conceptos básicos:

- Trigger: Evento que inicia el workflow (cron, webhook, email)
- Nodo: Cada paso del proceso (HTTP Request, IA, Gmail, Slack)
- Conexión: Pasa datos de un nodo al siguiente
- Credencial: Token o API key para autenticar servicios

Nodos de IA disponibles:

- AI Agent: Agente que razona y usa herramientas
- AI Chain: Secuencia de prompts encadenados
- AI Classifier: Clasifica texto en categorías
- AI Summarizer: Resume textos largos
- AI Text Generator: Genera texto con prompt personalizado

Lo que distingue a n8n en un entorno de trabajo real es que puedes alojarlo en tu propio servidor y que los datos pasan de nodo a nodo de forma visible: en cada paso puedes inspeccionar qué entró y qué salió, lo que facilita mucho depurar.

Un recurso que conviene conocer desde el principio son los flujos de error. Según la documentación de n8n, creas un workflow que empieza con el nodo Error Trigger y, en los ajustes del flujo que quieres vigilar, lo seleccionas como flujo de error. Cuando una ejecución automática falla, ese flujo recibe el nombre del workflow, el último nodo ejecutado y el mensaje de error, y puede avisarte por correo o mensajería. Un detalle práctico: este disparador no se activa en las pruebas manuales, solo cuando falla una ejecución automática, así que no concluyas que no funciona porque no salta mientras pruebas. Con un ejemplo inventado, si tu flujo de facturas falla un viernes por la noche porque caducó un token, te enterarás en minutos y no el lunes.

Claude Code hooks

Los hooks de Claude Code son scripts que se ejecutan automáticamente en momentos específicos:

Configuración en `.claude/settings.json`:

```
{
  "hooks": {
    "PreToolUse": [
      {
        "matcher": "Bash",
        "hooks": [
          { "type": "command", "command": "echo 'Verificando comando...'" }
        ]
      }
    ],
    "PostToolUse": [
      {
        "matcher": "Write|Edit",
        "hooks": [
          { "type": "command", "command": "jq -r '.tool_input.file_path' | xargs npx prettier"
        ]
      }
    ],
    "Notification": [
      {
        "matcher": "",
        "hooks": [
          { "type": "command", "command": "osascript -e 'display notification \"Claude Code ne"
        ]
      }
    ]
  }
}
```

El hook recibe los datos del evento como JSON por la entrada estándar (por ejemplo, `tool_input.file_path` con la ruta del fichero editado).

Hooks disponibles (entre otros):

- PreToolUse: antes de ejecutar una herramienta (puede bloquearla)
- PostToolUse: después de que una herramienta termine bien
- Notification: cuando Claude Code envía una notificación
- Stop: cuando Claude termina de responder

La documentación oficial describe con precisión cómo se comporta un hook de tipo comando. Claude Code le pasa la información del evento en formato JSON por la entrada estándar, incluido el nombre de la herramienta y sus parámetros. El código de salida decide lo que pasa después: salir sin error deja seguir el flujo, mientras que salir con el código dos en un evento previo a una herramienta bloquea esa acción y el mensaje de error se le devuelve a Claude como motivo. Así es como se construye, por ejemplo, un hook que impide borrados recursivos aunque alguien los pida.

Además de los cuatro eventos de la lista existen otros, como el inicio de sesión o el envío de cada mensaje del usuario. El ejemplo de arriba usa el formato actual, que agrupa bajo cada evento un filtro de herramientas y una lista de acciones con su tipo; si encuentras ejemplos antiguos con el

comando directamente bajo el filtro y variables como la ruta del fichero en el entorno, ya no sirven. Contrasta siempre con la documentación vigente. Y un aviso de seguridad: un hook ejecuta comandos con tus permisos, así que revisa los hooks de cualquier repositorio que clones antes de confiar en él.

Zapier y Make: sin código

Para automatizaciones sin código, Zapier y Make son las opciones más accesibles:

Workflow en Zapier (5 min):

Trigger: Nuevo email en Gmail con etiqueta "Lead"

Paso 1: Extraer nombre y email del remitente

Paso 2: Añadir a hoja de Google Sheets

Paso 3: Enviar mensaje de bienvenida con IA

Paso 4: Crear tarea en Notion

Paso 5: Notificar por Slack

Coste: Plan gratuito = 100 tareas/mes

Plan Starter = 750 tareas/mes (~19 USD)

En este tipo de herramientas, la lógica condicional es lo que convierte una cadena de pasos en un proceso útil. En Zapier, un filtro detiene la ejecución cuando los datos no cumplen una condición, y la herramienta llamada Paths reparte el flujo en ramas distintas según reglas que defines. Según la ayuda de Zapier, los pasos de filtro y de ramificación no consumen tareas; solo cuentan las acciones que realmente se ejecutan.

Un ejemplo con datos inventados: de cien correos etiquetados al mes, sesenta son respuestas automáticas. Un filtro al principio evita que esos sesenta recorran los cinco pasos siguientes. El error típico es poner la lógica al final, cuando ya se han gastado las acciones y, peor aún, cuando ya se ha enviado un mensaje de bienvenida a quien no debía recibirlo.

5 workflows que deberías automatizar

1. Procesamiento de leads: Email entrante, extraer datos, clasificar por prioridad con IA, añadir al CRM, notificar al comercial.

2. Curaduría de contenido: RSS de 10 fuentes, IA filtra y resume, genera post para redes sociales, programa publicación.

3. Informe semanal: Recopilar métricas de Analytics, Stripe, CRM. IA genera resumen ejecutivo. Enviar por email el lunes a las 8:00.

4. Onboarding de clientes: Nuevo pago en Stripe, crear cuenta, enviar secuencia de emails de bienvenida, agendar llamada de onboarding.

5. Monitoreo de marca: Buscar menciones en Google y redes sociales, IA clasifica sentimiento (positivo/negativo/neutro), alertar si hay crisis.

Errores al automatizar

- **Automatizar antes de entender:** Primero haz la tarea manualmente 10 veces. Luego automatiza.
- **No gestionar errores:** ¿Qué pasa si la API falla? ¿Si el email no se envía? Siempre pon manejo de errores.
- **Automatización invisible:** Si nadie sabe que existe, nadie la mantiene. Documenta tus workflows.
- **Sin alertas:** Configura notificaciones cuando un workflow falla. Un workflow roto que nadie detecta es peor que hacerlo manual.
- **Over-engineering:** Empieza simple. 3 pasos que funcionan valen más que 20 pasos que fallan.

Escalar automatizaciones

Cuando tus automatizaciones crecen:

- **Documenta cada workflow:** Qué hace, qué trigger lo activa, qué credenciales usa, quién lo mantiene.
- **Centraliza en una herramienta:** No tengas workflows en 5 plataformas diferentes.
- **Monitoriza costes:** Las APIs de IA cobran por uso. Un workflow mal configurado puede generar facturas inesperadas.
- **Revisión mensual:** ¿Siguen siendo útiles? ¿Se puede optimizar? ¿Hay nuevas herramientas mejores?

Tu primer workflow

No intentes automatizar todo a la vez. Elige UNA tarea que hagas todos los días y que te lleve más de 15 minutos. Automatízala esta semana. Luego pasa a la siguiente.

Caso práctico: workflow de curaduría de contenido en n8n

WORKFLOW: Curaduría diaria de noticias del sector

TRIGGER: Schedule (cada día a las 7:00)

PASO 1: RSS Feed Read

Fuentes: 10 blogs y medios del sector

Output: lista de artículos de las últimas 24h

PASO 2: AI Classifier (nodo IA)

Prompt: "De estos 30 artículos, selecciona los 3 más relevantes para CTOs de empresas tech en España.

Para cada uno: título, resumen en 2 frases, URL."

Modelo: GPT-4o o Claude Haiku

PASO 3: AI Text Generator (nodo IA)

Prompt: "Con estos 3 artículos, genera:

1. Un post para LinkedIn (150 palabras, 3 hashtags)
2. Un tweet (280 chars)
3. Un bloque para la newsletter semanal"

PASO 4: Google Sheets

Añadir los 3 artículos seleccionados al spreadsheet de contenido para tracking

PASO 5: Buffer / Postiz API

Programar el post de LinkedIn para las 9:00

Programar el tweet para las 12:00

PASO 6: Slack Notification

Enviar al canal #marketing: "Contenido del día listo.

Post LI: [preview]. Tweet: [preview]."

RESULTADO: 0 minutos/día (vs 45 minutos manual)

COSTE: ~0.02 USD/día en API de IA

Tabla: herramientas de automatización comparadas

CRITERIO	N8N	ZAPIER	MAKE	CLAUDE HOOKS
Precio (entrada)	Gratis (self-hosted)	Gratis (100 tasks/mes)	Gratis (1.000 ops/mes)	Incluido en Claude
Self-hosted	Sí	No	No	Sí (local)
Nodos de IA	Sí (nativos)	Sí (limitados)	Sí (limitados)	Claude directo
Complejidad	Media-alta	Baja	Media	Baja (solo dev)
Integraciones	400+ nativos + HTTP	6.000+	1.500+	Scripts locales
Mejor para	Dev/tech, flujos complejos	No-code, simple	Visual, coste medio	Desarrollo, pre/post tool

Claude Code hooks: ejemplo avanzado

```
# Configuración en .claude/settings.json

{
  "hooks": {
    "PreToolUse": [
      {
        "matcher": "Bash",
        "hooks": [
          { "type": "command", "command": "echo 'Verificando que no es destructivo...'" }
        ]
      }
    ],
    "PostToolUse": [
      {
        "matcher": "Write|Edit",
        "hooks": [
          { "type": "command", "command": ".claude/hooks/formatear.sh" }
        ]
      }
    ],
    "Notification": [
      {
        "matcher": "",
        "hooks": [
          { "type": "command", "command": "jq -r '.message' | xargs -I{} terminal-notifier -ti{}" }
        ]
      }
    ]
  }
}

# .claude/hooks/formatear.sh (recibe el evento como JSON por la entrada estándar)
#!/bin/bash
f=$(jq -r '.tool_input.file_path')
case "$f" in
  *.py) ruff check "$f" --fix 2>/dev/null || true ;;
  *.js|*.ts|*.tsx|*.json|*.css) npx prettier --write "$f" 2>/dev/null || true ;;
esac

# Resultado: cada archivo que Claude escribe o edita se formatea
# automáticamente con Prettier (JS/TS) o Ruff (Python).
# Recibes una notificación nativa cuando Claude Code te necesita.
```

Workflow para principiantes: paso a paso en Zapier

WORKFLOW: Lead scoring automático

1. TRIGGER: Nuevo formulario en tu web (Typeform/Google Forms)
→ Zapier detecta la nueva respuesta

2. PASO 1: Extraer datos
→ Nombre, email, empresa, cargo, mensaje

3. PASO 2: Enrichment con IA
→ Enviar a ChatGPT: "Clasifica este lead del 1 al 10 basado en: cargo (CEO/CTO = alto), tamaño de empresa (si se menciona), urgencia del mensaje."

4. PASO 3: Añadir a Google Sheets
→ Con columnas: nombre, email, empresa, score, fecha

5. PASO 4: Condicional
→ Si score $\geq$ 7: enviar email personalizado inmediato
→ Si score 4-6: añadir a secuencia de nurture
→ Si score < 4: solo guardar en spreadsheet

6. PASO 5: Notificar por Slack
→ Canal #ventas: "Nuevo lead: [nombre] de [empresa], score: [X]"

TIEMPO DE SETUP: 30 minutos

RESULTADO: Cada lead clasificado y enrutado en <1 minuto (vs 15 min manual)

Errores comunes al automatizar

- **Automatizar procesos rotos:** Si tu proceso manual es ineficiente, automatizarlo produce ineficiencia a escala. Primero optimiza el proceso, luego automatiza.
- **No contemplar fallos:** La API de IA puede fallar, el servidor puede caer, los datos pueden venir mal formateados. Siempre añade nodos de error handling y alertas de fallo.
- **Workflows sin documentación:** Un workflow que solo tú entiendes es un riesgo operativo. Documenta: qué hace, qué trigger lo activa, qué credenciales usa y quién es responsable de mantenerlo.
- **No monitorizar costes:** Un workflow con nodo de IA que se dispara 1.000 veces al día puede generar facturas inesperadas. Pon límites y alertas de coste.
- **Automatizar sin probar manualmente primero:** Haz la tarea manualmente al menos 10 veces. Identifica variaciones y excepciones. Luego automatiza el caso base y maneja las excepciones con lógica condicional.

Consejo: El mejor workflow es el que eliminas, no el que creas. Antes de automatizar, pregúntate: ¿esta tarea es realmente necesaria? A veces la mejor automatización es dejar de hacer algo que no aporta valor.

IAcademy · iacademy.com · © 2026 Riskitera S.L.U.

MT09: Workflow automatizado · Experto

¿Dudas? Usa el tutor IA en iacademy.com/course/mt09/