Jano.

Debugging en n8n: los niveles, las trampas y los atajos

Introducción

Con esta hoja tenés a mano el método del video: qué nivel usar según el caso, las trampas que esconden los errores y los atajos que ahorran tiempo. Al final están los pasos para armar la práctica con los flujos de la clase.

Importante

Una ejecución en verde solo quiere decir que ningún error la frenó. No quiere decir que hizo lo que tenía que hacer. Revisá también las verdes (Sección 2).

Sección 1: Los niveles y cuándo usar cada uno

1. Nivel 4 · Claude entra a tu n8n

Cuándo: siempre que puedas crear una API key. Es el que más tiempo te ahorra: Claude revisa todas las ejecuciones, encuentra la causa, arregla, prueba y recupera lo perdido.

1. En n8n: **Settings** → **n8n API** → **Create an API key**. Copiala: n8n te la muestra una sola vez.
2. Descomprimí **n8n-con-claude.zip**. Abrí la app de Claude (claude.ai/download; necesita un plan pago, el Pro alcanza), pestaña **Code**, y abrí una sesión nueva en esa carpeta.
3. En la terminal de la sesión corré python3 conectar.py (en Windows, python conectar.py). Te pide la dirección de tu n8n y la API key, y la guarda en el archivo .env. Claude nunca la ve.
4. En el chat de la sesión, pedile lo que necesites. Pedidos que sirven:

   ```
   revisá las ejecuciones de hoy de [flujo]: qué falló y qué quedó en verde pero salió mal
   explicame qué pasó en la ejecución [número], nodo por nodo
   arreglalo y probalo con [un pedido de ejemplo]
   las ejecuciones que fallaron por esto, reintentalas
   las que quedaron en verde pero no se guardaron, recuperalas llamando de nuevo al webhook
   revisá este JSON antes de que lo pegue: [archivo]
   ```

5. Para que la revisión corra sola todos los días, programala en tu computadora (en Mac o Linux, con crontab -e; en Windows, con el Programador de tareas). Esta línea revisa las últimas 24 horas a las 9 y deja el reporte en la carpeta **reportes**:

   ```
   0 9 * * * cd /ruta/a/n8n-con-claude && python3 n8n.py revisar --horas 24 > reportes/$(date +\%F).txt
   ```

Cambiá /ruta/a/ por la carpeta donde la descomprimiste y creá antes la carpeta **reportes**.

2. Nivel 3 · Claude con el JSON

Cuándo: no tenés API key (el n8n de un cliente donde no sos el dueño, un plan sin API) o querés una consulta rápida. Alcanza con Claude en el navegador, aunque sea el plan gratis.

1. Seleccioná los nodos que fallan y tocá **Cmd + C** (Ctrl + C en Windows). Para el flujo entero: **Cmd + A** y **Cmd + C**, o el menú de los tres puntos → **Download**.
2. Pegalo en Claude con el síntoma: qué pediste, qué pasó y qué tenía que pasar. Pedile los nodos corregidos en JSON para pegar en n8n, con nodes y connections: sin eso, n8n no lo pega y avisa "Problem importing workflow".
3. Borrá primero los nodos viejos: así los nuevos entran con el mismo nombre y las expresiones de los otros nodos, que los llaman por su nombre, siguen andando. Pegá con **Cmd + V**, conectalos y revisá lo de la Sección 3. Después, **Publish**.

No le mandes capturas: con una captura la IA ve lo que ves vos; con el JSON ve lo que n8n tiene adentro (cada expresión, cada opción, la configuración de errores).

3. Nivel 1 · La ejecución por dentro

Cuándo: querés ver un caso puntual con tus ojos, un cliente te reclama un pedido, o querés entender qué hizo Claude.

1. En el workflow, **Executions**. Filtrá por **Status** para ver solo las que fallaron.
2. Leé el error en tres partes: qué pasó, en qué nodo o condición y con qué item.
3. Abrí la **entrada** del nodo que se queja: muchas veces el error nace antes.
4. Si el flujo llama a un subflujo, mirá también las ejecuciones del subflujo.
5. **Debug in editor** trae los datos de una ejecución fallida al editor, para repetirla con el pedido real (en una exitosa, el botón se llama **Copy to editor**). Necesita la licencia gratis del video 1. Cuando termines, soltá los datos fijados con **P**.
6. Si el error hizo perder datos, después del arreglo reintentá las ejecuciones fallidas: **Retry with currently saved workflow** las repite desde el nodo que falló, con tu versión arreglada. Las ejecuciones en verde no se pueden reintentar: si el flujo arranca con un webhook, recuperalas llamándolo de nuevo con el mismo pedido (en el nivel 4, se lo pedís a Claude).
7. Para encontrar la ejecución de un cliente entre cientos: sumá al flujo un nodo **Execution Data** que guarde, por ejemplo, el número de pedido, y buscalo con el filtro **Highlighted data** (pide el registro gratis de n8n).

Sección 2: Verde no es bien

Una ejecución puede terminar en verde y haber salido mal. Estas son las causas que más se repiten:

1. **Un nodo con On Error en Continue que falló.** El flujo siguió como si nada, y en la ejecución ese nodo se ve con el check verde: el error está adentro de su salida. Usalo solo si de verdad querés que siga, y nunca para esconder un error.
2. **Un subflujo que falló.** Su error queda en las ejecuciones del subflujo, no en las del flujo que lo llamó.
3. **Una herramienta de un agente que dio error.** El agente responde igual, con lo que tenga.
4. **Un nodo que devolvió 0 items.** n8n corta el flujo ahí, sin error. Si tiene que seguir

igual: **Settings** del nodo → **Always Output Data**.

5. **Un webhook que respondió vacío.** Si el flujo usa Respond to Webhook y se corta antes, quien llamó recibe un 200 sin nada.
6. **Un resultado distinto del esperado.** Un total mal calculado no da error. Para eso están las pruebas del video 11.

Sección 3: Lo que no viaja en el JSON

Cuando pasás nodos de un n8n a otro, o pegás lo que te devuelve Claude:

1. **Credenciales:** viajan con su id y su nombre, nunca con la clave. Si ese id no existe en tu n8n (o Claude puso solo el nombre), al pegar n8n la saca y elige sola la última credencial de ese tipo que editaste, con un aviso. Revisala siempre: abrí el nodo y elegí la credencial correcta de la lista, sobre todo si tenés una por cliente. Si Claude escribe un nodo nuevo, pedile que copie la credencial de un nodo tuyo que ya la use.
2. **Subflujos:** van por id. Importá primero el subflujo y después, en el nodo que lo llama, elegilo de la lista. En producción n8n llama a la versión **publicada** del subflujo: publicalo antes que el workflow principal.
3. **Hojas de Google Sheets:** elegilas de la lista (quedan por id). Escritas por nombre, se rompen cuando alguien las renombra.
4. **Expresiones:** en el JSON empiezan con =, como "={{ $json.total }}". Sin el =, n8n manda el texto tal cual.
5. **Datos fijados:** lo que copiás lleva los datos fijados de esos nodos. Si son de un cliente, soltalos antes de compartir el flujo.
6. **Workflow entero:** si pegás un workflow completo con Cmd + V en uno abierto, n8n le cambia el nombre al que tenés abierto. Pegalo en un workflow nuevo o usá Import from file.

Sección 4: Atajos

1. **Cmd + Enter:** ejecutar el workflow.
2. **P:** fijar o soltar los datos del nodo elegido.
3. **D:** desactivar o activar el nodo.
4. **F2:** renombrar el nodo.
5. **N:** agregar un nodo.
6. **L:** abrir los Logs, con cada nodo y lo que tardó.
7. **Cmd + D:** duplicar el nodo.
8. **Shift + Alt + T:** ordenar el workflow.
9. **=** en un campo vacío: pasar a expresión y ver el resultado mientras escribís.

En Windows, Cmd es Ctrl.

Sección 5: Para armar la práctica

1. Creá una planilla de Google **Ferretería El Tornillo** con dos pestañas: **Lista de precios** (importá lista-de-precios.csv) y **Cotizaciones**, con el encabezado fecha, producto, cantidad, moneda, total en la fila 1, en minúscula y uno por columna: si los títulos no coinciden, el registro falla con "Column names were updated after the node's setup". En la fila de la llave inglesa, escribí el stock como '8 (con apóstrofo): queda como texto, como lo dejaría alguien del negocio.
2. Importá los flujos en este orden: **guardar-cotizacion.json**, **cotizador.json** y

simular-pedidos.json (menú de los tres puntos → Import from file).
3. En cada nodo de Google Sheets, revisá la credencial que eligió n8n y elegí tu planilla. En el nodo **Guardar**, elegí el subflujo "Guardar cotización" de la lista. En **Enviar al Cotizador**, poné la dirección de tu n8n.
4. Publicá primero **Guardar cotización** y después el **Cotizador**. Corré Simular pedidos.

Sección 6: Si algo sale mal

1. Cada vez que Claude cambia un workflow, el programa guarda una copia en la carpeta **backups**. Pedile a Claude que vuelva a subir la última.
2. n8n también guarda las versiones de las últimas 24 horas: menú de los tres puntos → **Version history** → **Restore version**, y después **Publish**.
3. Si Claude cambió un flujo que tenías abierto en el navegador, recargá la página antes de tocar nada: si tocás algo en la pestaña vieja, n8n guarda esa versión encima del arreglo sin avisarte.
4. Si la API key se filtró, borrala en **Settings** → **n8n API**, creá otra y corré python3 conectar.py.

Clases filtradas de n8n · Jano Llosa · @one.jano · jano.my. Usalo libre en tus proyectos y con tus clientes; si lo compartís, nombrá la fuente.