

Chuleta de n8n: expresiones, datos y errores típicos

Todo lo dinámico en n8n va entre `{{ }}`. Dentro tienes cuatro variables (`$json`, `$node`, `$now`, `$env`) y una trampa que se lleva por delante a casi todo el mundo: **los datos del webhook no están en la raíz, están bajo `.body`**.



[!\[\]\(666e09182d4cd268646ea700ea60dcdf_img.jpg\) Descargar en PDF \(gratis, sin registro\)](#)

Variables y sintaxis básica

Qué escribes	Qué hace	Cuándo usarlo
<code>{{ }}</code>	Marca lo que n8n tiene que evaluar. Sin ellas, es texto literal.	Siempre que el valor dependa de los datos. Nunca las anides.
<code>\$json.campo</code>	Un campo de la salida del nodo anterior.	El caso normal: trabajar con lo que acaba de llegar.
<code>\$node["Nombre"].json.campo</code>	Un campo de <i>cualquier</i> nodo anterior por su nombre.	Cuando necesitas algo de dos o tres pasos atrás, no del inmediato.
<code>\$now</code>	La fecha y hora actuales.	Sellos de tiempo, nombres de fichero, filtros por fecha.
<code>\$env.CLAVE</code>	Una variable de entorno de la instancia.	Con cuidado: muchas instancias lo bloquean. Para secretos, usa credenciales.
<code>\$json['con espacios']</code>	Acceso por corchetes.	Obligatorio si la clave lleva espacios, acentos o símbolos.

Dónde están tus datos

El 90% de los "no me llega el dato" se resuelve en esta tabla.

De dónde viene	Cómo se accede	La trampa
Nodo Webhook	<code>{{ \$json.body.email }}</code>	El error nº1. El nodo envuelve los datos bajo <code>.body</code> para conservar cabeceras, params y query. En la raíz no hay nada tuyo.
Campo anidado	<code>{{ \$json.user.email }}</code>	Si el padre no existe, el error dirá "cannot read property of undefined". Revisa la ruta entera, no la última clave.
Elemento de un array	<code>{{ \$json.items[0].name }}</code>	Los índices empiezan en 0. El último: <code>{{ \$json.items[\$json.items.length - 1].name }}</code> .

De dónde viene	Cómo se accede	La trampa
Otro nodo, con espacios en el nombre	<code>{{ \$node["HTTP Request"].json.data }}</code>	Distingue mayúsculas. "http request" no encuentra nada.
Webhook desde otro nodo	<code>{{ \$node["Webhook"].json.body.email }}</code>	El <code>.body</code> sigue ahí aunque referencias el nodo por su nombre.
Cuántos elementos hay	<code>{{ \$json.users.length }}</code>	Sin paréntesis: es una propiedad, no un método.

Transformar el dato al vuelo

Qué escribes	Qué hace	Cuándo usarlo
<code>\$now.toFormat('yyyy-MM-dd')</code>	Formatea la fecha (por debajo es Luxon).	Nombres de fichero, campos de fecha, agrupar por día.
<code>\$now.plus({days: 7})</code>	Suma tiempo. minus resta.	Vencimientos, recordatorios, ventanas de fechas.
<code>\$json.email.toLowerCase()</code>	Normaliza texto. También trim, replace, substring.	Antes de comparar o de usar el valor como clave.
<code>\$json.tags.split(',').join(', ')</code>	Parte y vuelve a unir.	Para convertir listas separadas por comas en algo legible.
<code>\$json.precio * 1.21</code>	Aritmética directa.	IVA, descuentos, totales. Sin nodo extra.
<code>\$json.email 'sin-email@ejemplo.com'</code>	Valor por defecto si viene vacío.	Para que un campo ausente no reviente el resto del flujo.
<code>\$json.estado === 'activo' ? 'Sí' : 'No'</code>	Condicional en una línea.	Textos que cambian según un campo, sin montar una rama entera.

Dónde NO van las expresiones

Sitio	Qué pasa si lo intentas	Qué hacer
Nodo Code	Escribes JavaScript, no expresiones. <code>'={{ \$json.email }}'</code> es una cadena de texto literal, no el dato.	Acceso directo: <code>\$json.email</code> , o <code>\$input.all()</code> para todos los elementos.
Ruta del webhook	No se evalúa. La ruta tiene que ser estática.	Un texto fijo, tipo formulario-contacto.
Campos de credenciales	No se evalúan y, si funcionara, dejarías el secreto a la vista.	El sistema de credenciales de n8n, para eso está.

Errores típicos

- **Buscar los datos del webhook en la raíz.** Es el error más repetido de n8n con diferencia: `$json.name` viene vacío porque el dato está en `$json.body.name`. Si algo del webhook "no llega", empieza siempre por ahí.
- **Olvidar las llaves.** `$json.email` a secas se guarda como texto literal, sin avisar de nada. Si en la vista previa ves el nombre de la variable en vez del valor, es esto.

- **Usar expresiones dentro de un nodo Code.** Dentro de Code es JavaScript normal: `$json.email`, sin llaves ni comillas. Poner `'={{...}}'` te guarda la cadena, no el dato.
- **Escribir mal el nombre del nodo.** Distingue mayúsculas y tiene que coincidir con el lienzo. Y si lleva espacios, van corchetes y comillas sí o sí: `$node["HTTP Request"]`.
- **Dar por hecho que \$env funciona.** Muchas instancias tienen bloqueado el acceso a variables de entorno desde los nodos. Si te da error, no es tu expresión: usa credenciales o pasa el valor por un nodo Set.

Preguntas frecuentes

¿Por qué mi expresión sale como texto literal?

Porque le faltan las llaves dobles. n8n solo evalúa lo que va entre `{{` y `}}`; cualquier otra cosa la trata como texto. Ojo también con las llaves triples: envolver dos veces tampoco funciona.

El webhook recibe los datos pero el campo llega vacío, ¿por qué?

Es el error más repetido de n8n. El nodo Webhook no deja los datos en la raíz: los mete bajo `.body`, para no pisar las cabeceras, los parámetros y la query. Donde escribiste `$json.email` tiene que poner `$json.body.email`.

¿Puedo usar expresiones dentro de un nodo Code?

No, y es una confusión habitual. Dentro de un nodo Code escribes JavaScript normal y accedes directamente: `$json.email`, sin llaves y sin comillas. Las llaves solo valen en los campos de los nodos, no dentro del código.

¿Cómo referencio un nodo cuyo nombre lleva espacios?

Con corchetes y comillas: `$node["HTTP Request"].json.data`. El nombre tiene que coincidir exactamente con el del lienzo, mayúsculas incluidas: `"http request"` no encuentra el nodo.

¿Cómo pongo la fecha de hoy con el formato que quiero?

Con `$now` y `toFormat`: `$now.toFormat('yyyy-MM-dd')`. Para sumar o restar tiempo, `plus` y `minus` con un objeto: `$now.plus({days: 7})`. Por debajo es Luxon, así que sus formatos valen.

Vídeos relacionados

- [Crea tu PRIMER flujo de automatización con n8n](#) — de cero a un flujo funcionando.
- [Cómo crear agentes de IA en 2026 \(sin programación\)](#)
- [Deja de crear agentes de IA, haz esto en su lugar](#)
- [El error fatal al crear agentes de IA](#) — el que te puede costar los datos.

¿Quieres pasar de flujos sueltos a agentes que trabajan solos? El curso de n8n de Frogames va paso a paso y con proyectos reales.

[Ver los cursos de Frogames →](#)